



(12) 发明专利申请

(10) 申请公布号 CN 102193862 A

(43) 申请公布日 2011. 09. 21

(21) 申请号 201110065878. X

(22) 申请日 2011. 03. 09

(30) 优先权数据

12/720, 691 2010. 03. 10 US

(71) 申请人 微软公司

地址 美国华盛顿州

(72) 发明人 J·A·穆哈斯凯 R·沃格里内克

B·S·沃德斯沃斯

(74) 专利代理机构 上海专利商标事务所有限公

司 31100

代理人 钱静芳

(51) Int. Cl.

G06F 11/36 (2006. 01)

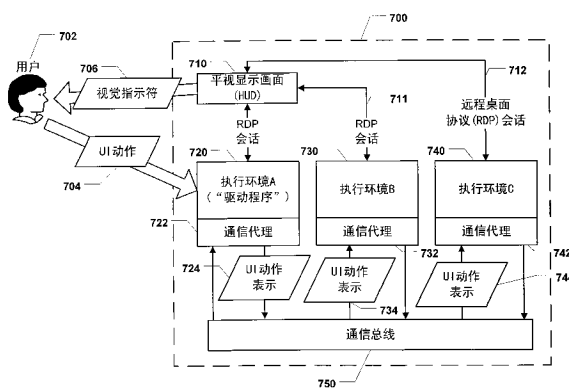
权利要求书 2 页 说明书 8 页 附图 6 页

(54) 发明名称

在多个执行环境中测试用户界面

(57) 摘要

公开了在多个执行环境中测试用户界面的方法,系统和计算机可读介质。一种具体方法包括:选择一个或多个 UI 测试以及其中要运行该 UI 测试的一个或多个执行环境。各执行环境中的一个被指定为驱动程序执行环境。显示与该驱动程序执行环境相对应的驱动程序 UI。当在驱动程序 UI 处接收到 UI 动作时,将该 UI 动作的数据表示从驱动程序执行环境发送到每一个其他执行环境。该 UI 动作在每一个其他执行环境中基本上同时重复。



1. 一种计算机实现的方法,包括:
选择与基于用户界面 (UI) 的应用程序相关联的一个或多个测试 (402);
选择多个执行环境 (404),其中所述多个执行环境中的一个被指定为驱动程序执行环境 (120);
显示与所述驱动程序执行环境相对应的驱动程序 UI (408);
在所述驱动程序 UI 处接收与所述一个或多个测试相关联的 UI 动作 (410);以及
将所述 UI 动作的表示 (124) 从所述驱动程序执行环境 (120) 发送 (414) 到每一个其他执行环境 (130、140),其中所述 UI 动作 (124) 基本上同时在每一个其他执行环境 (130、140) 处重复。
2. 如权利要求 1 所述的计算机实现的方法,其特征在于,还包括:在所述驱动程序 UI 处接收第二 UI 动作,并将所述第二 UI 动作的表示从所述驱动程序执行环境发送到每一个其他执行环境。
3. 如权利要求 1 所述的计算机实现的方法,其特征在于,发送所述 UI 动作的表示包括:将所述 UI 动作转换成一组输入设备控制并发送该组输入设备控制。
4. 如权利要求 3 所述的计算机实现的方法,其特征在于,所述一组输入设备控制包括键盘键入、鼠标移动、鼠标点击、触笔控制、触摸屏控制、多点触摸控制、或以上的任何组合。
5. 如权利要求 3 所述的计算机实现的方法,其特征在于,所述一组输入设备控制以串行化格式发送。
6. 如权利要求 1 所述的计算机实现的方法,其特征在于,发送所述 UI 动作的表示包括:通过通信总线来广播所述 UI 动作的表示,所述通信总线耦合到所述多个执行环境中的每一个。
7. 如权利要求 1 所述的计算机实现的方法,其特征在于,所述驱动程序 UI 在平视显示画面 (HUD) 处显示,所述平视显示画面被配置为显示所述多个执行环境中的每一个。
8. 如权利要求 7 所述的计算机实现的方法,其特征在于,还包括:在所述 HUD 处显示视觉指示符,所述视觉指示符指示第一执行环境具有第一状态,该第一状态不同于第二执行环境的第二状态。
9. 如权利要求 8 所述的计算机实现的方法,其特征在于,所述第一状态或第二状态中的至少一个包括:UI 屏幕截图、导航状态、模型状态、自动化状态、参数状态、一个或多个性能度量、或以上的任何组合。
10. 如权利要求 8 所述的计算机实现的方法,其特征在于,还包括:在日志文件中创建条目,以指示所述第一执行环境的第一状态和所述第二执行环境的第二状态之间的差别,并且所述 HUD 经由与所述至少一个执行环境的远程桌面协议 (RDP) 会话来显示至少一个执行环境。
11. 一种计算机系统,包括:
存储器 (630);以及
耦合到所述存储器 (630) 的处理器 (620),其中所述处理器 (620) 被配置为执行引起用户界面 (UI) 测试应用程序的执行的指令,所述用户界面测试应用程序包括:
平视显示画面 (HUD) (110),被配置为:
显示多个执行环境 (120、130、140) 中的每一个,其中所述多个执行环境中的一个被指

定为驱动程序执行环境；

在所述驱动程序执行环境处接收与 UI 测试相关联的 UI 动作 (104)；以及

将所述 UI 动作的表示 (124、134、144) 从所述驱动程序执行环境 (120) 发送到每一个其他执行环境 (130、140)，其中所述 UI 动作 (104) 基本上同时在每一个其他执行环境 (130、140) 处重复；以及

通信总线 (150)，所述通信总线耦合到所述多个执行环境 (120、130、140) 中的每一个，并且被配置为从所述驱动程序执行环境 (120) 向每一个其他执行环境 (130、140) 广播数据。

12. 如权利要求 11 所述的计算机系统，其特征在于，所述 UI 测试应用程序进一步包括：状态比较器，被配置成将第一执行环境的第一状态与第二执行环境的第二状态进行比较。

13. 如权利要求 12 所述的计算机系统，其特征在于，所述 HUD 进一步被配置为当所述状态比较器检测到两个执行环境之间的状态失配时显示视觉指示符。

14. 如权利要求 11 所述的计算机系统，其特征在于，所述多个执行环境中的至少一个进一步包括：测试记录器，被配置为存储多个 UI 动作的表示。

15. 如权利要求 14 所述的计算机系统，其特征在于，所述多个执行环境中的至少一个进一步包括：测试播放器，被配置为重现所述多个 UI 动作。

在多个执行环境中测试用户界面

技术领域

[0001] 本发明涉及应用程序的测试,尤其涉及多个执行环境中的应用程序测试。

背景技术

[0002] 软件厂商经常在多个计算平台上发布软件应用程序。在发布之前,经常在每个计算平台上测试该软件应用程序。在多个平台上反复的软件测试是耗时的。例如,每次对特定平台的测试重复由于为每次测试重复而反复重配置应用程序,可能引起时间和资源上的开销。

发明内容

[0003] 公开了一种在多个执行环境中测试用户界面 (UI) 的方法。在一个执行环境 (例如,“驱动程序”执行环境) 中进行的 UI 动作在一个或多个其他执行环境中自动且基本上同时重复。可向用户 (例如,UI 测试者) 提供平视显示画面 (HUD), 该 HUD 包括由每个执行环境所生成的 UI 并且标识这些执行环境之间的状态或外观上的不同之处。

[0004] 提供本概述是为了以简化的形式介绍将在以下详细描述中进一步描述的一些概念。本概述并不旨在标识出所要求保护的主题的关键特征或必要特征,也不旨在用于限定所要求保护的主题的范围。

附图说明

[0005] 图 1 是示出在多个执行环境中测试用户界面的系统的具体实施例的示图。

[0006] 图 2 是示出在多个执行环境中测试用户界面的系统的另一个具体实施例的示图。

[0007] 图 3 是示出图 1 的系统或图 2 的系统上的数据流的具体实施例的数据流程图。

[0008] 图 4 是示出在多个执行环境中测试用户界面的方法的具体实施例的流程图。

[0009] 图 5 是示出显示驱动程序执行环境和其他执行环境的平视显示画面 (HUD) 的具体实施例的屏幕截图。

[0010] 图 6 是一计算环境的框图,该计算环境包括可用于支持如图 1-5 所示的计算机实现的方法、计算机程序产品以及系统组件的各实施例的计算设备。

具体实施方式

[0011] 在一具体实施例中,一种计算机实现的方法包括:选择与基于用户界面 (UI) 的应用程序相关联的一个或多个测试,并且选择多个执行环境。多个执行环境中的一个被指定为驱动程序执行环境。该方法还包括:显示与驱动程序执行环境相对应的驱动程序 UI。该方法进一步包括:在驱动程序 UI 处接收与一个或多个测试相关联的 UI 动作。该方法包括:将 UI 动作的表示从驱动程序执行环境发送到每一个其他执行环境。该 UI 动作在每一个其他执行环境中基本上同时重复。在一个替换实施例中,可以在驱动程序执行环境中进行随机 (ad hoc) 测试,并且该随机测试能在其他执行环境中复制。

[0012] 在另一个具体实施例中,一种计算机系统包括存储器和耦合到该存储器的处理器。该处理器被配置为执行指令,该指令导致用户界面 (UI) 测试应用程序,该用户界面测试应用程序包括平视显示画面 (HUD) 和通信总线。该 HUD 被配置为显示多个执行环境中的每一个,其中多个执行环境中的一个被指定为驱动程序执行环境。该 HUD 被配置为在驱动程序执行环境处接收与 UI 测试相关联的 UI 动作。该 HUD 进一步被配置为将 UI 动作的表示从驱动程序执行环境发送到每一个其他执行环境。该 UI 动作在每一个其他执行环境中基本上同时重复。通信总线耦合到多个执行环境中的每一个,并被配置为从驱动程序执行环境向每一个其他执行环境广播数据。

[0013] 在另一个具体实施例中,一种计算机可读介质包括指令,该指令当被计算机执行时使得计算机选择与基于用户界面 (UI) 的应用程序相关联的一个或多个测试,并选择多个执行环境。多个执行环境中的一个被指定为驱动程序执行环境。该指令还使得计算机初始化在多个执行环境中的每一个处的通信代理,并显示与驱动程序执行环境相对应的驱动程序 UI。该指令进一步使得计算机在驱动程序 UI 处接收与一个或多个测试相关联的 UI 动作。该指令使得计算机将 UI 动作的表示从驱动程序执行环境处的通信代理通过通信总线发送到每一个其他执行环境处的通信代理。该 UI 动作在每一个其他执行环境中基本上同时重复。

[0014] 图 1 示出了在多个执行环境中测试用户界面 (UI) 的系统 100 的具体实施例。系统 100 包括平视显示画面 (HUD) 110 和通信总线 150。该系统还包括多个执行环境 (例如,说明性执行环境 120、130 和 140)。多个执行环境中的一个被指定为驱动程序执行环境。例如,在图 1 示出的具体实施例中,“执行环境 A”被指定为驱动程序执行环境 120。在一具体实施例中,系统 100 由计算设备来实现。一般而言,图 1 的系统 100 可操作为在执行环境 120、130 和 140 的每一个处基本上同时测试 UI。例如,UI 测试可以是随机测试或可基于预定测试例和场景。

[0015] 驱动程序执行环境 120 可被配置为在执行环境 120、130 和 140 的同时 UI 测试期间从用户 102 接收 UI 动作 104。驱动程序执行环境 120 (或驱动程序执行环境 120 处执行的测试应用程序) 可将 UI 动作 104 转换成 UI 动作表示 124。例如,UI 动作表示 124 可包括一组输入设备控制 (例如,键盘键入、鼠标移动、以及鼠标点击、触笔控制、触摸屏控制、多点触摸控制、或任何其他输入设备控制)。输入设备控制也可包括定时参数 (例如,等待时间)。UI 动作表示 124 还可包括自动生成的软件代码,该软件代码可执行来重复 UI 动作 104。驱动程序执行环境 120 可将 UI 动作表示 124 (例如,以串行化的格式) 发送到每一个其他执行环境 130、140。在一具体实施例中,UI 动作表示 124 由驱动程序执行环境 120 的通信代理 122 通过通信总线 150 广播到其他执行环境 130、140 的通信代理 132、142。

[0016] 每一个其他执行环境 130 和 140 可接收 UI 动作表示 134、144,并基于 UI 动作表示 134、144 基本上同时重复 UI 动作 104。因此,UI 动作 104 (诸如,与 UI 测试相关联的 UI 动作) 可以在一个执行环境处执行,并可在多个其他执行环境中基本上同时重复。应注意,UI 动作表示 124、134 和 144 可以是相同的,并且 UI 动作表示 124、134 和 144 可以是机器及计算环境无关的。因此,UI 动作表示可以被发送到多个执行环境,而不执行 UI 动作表示的个别格式化或配置。

[0017] HUD 110 可显示每一个执行环境 120、130 和 140。例如,HUD 110 可通过显示设备

向用户 102 显示执行环境 120、130 和 140。在一具体实施例中,HUD 110 可通过远程桌面协议 (RDP) 会话来显示一个或多个执行环境。例如,HUD 110 可通过分别与非驱动程序执行环境 130 和 140 的 RDP 会话 111 和 112 来显示非驱动程序执行环境 130 和 140。在一具体实施例中,HUD 也可通过 RDP 会话来显示驱动程序执行环境 120。

[0018] 在一具体实施例中,HUD 110 还被配置为从用户 102 接收对新的驱动程序执行环境的指定。HUD 110 也可被配置为向用户 102 提供 UI 测试结果的视觉指示符(诸如,说明性视觉指示符 106)。例如,视觉指示符 106 可指示在每一个执行环境 120、130 和 140 处已经进行了 UI 动作 104 之后,第一执行环境具有不同于第二执行环境的状态。该状态可包括 UI 屏幕截图、导航状态、模型状态、自动化状态、参数状态、以及性能度量(例如,在 UI 动作的执行期间耗费的存储器和处理器时间量)中的一个或多个。将两个执行环境之间的状态进行比较可包括图像的逐位比较、带有投影的分辨率差别的图像的比较、出于其他辅助选项要求(例如,色盲,等等)对图像的调节、不同语言的串的比较(例如,当主测试操作者(例如用户 102(只说英语时,该比较可包括翻译以使得该说英语者能有效地测试其他语言)、不同显示介质的比较(例如,台式大小的对比移动设备大小的)(包括对屏幕分辨率差别的补偿),以及其他比较。

[0019] HUD 110 可比较并检测各种执行环境 120、130 和 140 的状态中的失配。HUD 进一步参考图 5 来描述并示出。用户 102 可基于视觉指示符 106 来采取行动(例如,报告隐错)。例如,为了使偏离的(例如,失配的)执行环境与其他执行环境保持一致,用户 102 可指定新的驱动程序执行环境,禁用系统 100 处的 UI 动作复制,并在偏离的执行环境处视需要做出 UI 动作。

[0020] 在一具体实施例中,当系统包括多个执行环境时,这些执行环境中的一个或多个可由虚拟机来实现。例如,驱动程序执行环境 120 可以是系统 100 的本机(诸如,“主机”)执行环境,而其他执行环境 130、140 可在系统 100 处由虚拟机来执行(诸如,作为“客”环境)。

[0021] 在操作中,可选择针对基于 UI 的应用程序的一个或多个测试以便在多个所选择的执行环境(例如,执行环境 120、130 和 140)中的每一个处执行。执行环境 120、130 和 140 可以在显示分辨率、文本语言(例如,英语对比西班牙语)、软件应用程序(例如,不同的 web 浏览器或同一个 web 浏览器的不同版本)、操作系统、硬件架构(例如,32 位对比 64 位)、机器状况(诸如,“崭新的”机器对比带有“进化状态的”机器)、在其他特性方面、或在以上各个的组合方面可以互相不同。应注意,这些不同仅仅是说明性的,且不可被认为是限制。HUD 110 可向用户 102 显示驱动程序执行环境 120,且驱动程序执行环境 120 可从用户接收 UI 动作 104。例如,UI 动作 104 可以是经由鼠标点击在 web 浏览器处的按钮按下。驱动程序执行环境 120 可基于 UI 动作 104 来生成 UI 动作表示 124,并将 UI 动作表示 124 发送到其他执行环境 130 和 140。

[0022] 其他执行环境 130 和 140 可基本上同时重复 UI 动作 104。例如,其他执行环境 130 和 140 可基本上同时执行 web 浏览器按钮按下。HUD 110 可显示各种执行环境 120、130、140。HUD 110 还可在检测到执行环境 120、130、140 中的两个之间的状态失配时显示一视觉指示符 106。例如,视觉指示符 106 可指示在 web 浏览器按钮按下之后,第三执行环境 140 具有不同于驱动程序执行环境 120 以及第二执行环境 130 的 UI 状态,由此指示了第三执行

环境 140 处的 web 浏览器的可能隐错。

[0023] 该测试过程可对在系统 100 处所接收的每一个 UI 动作 104 重复。例如,可以在驱动程序执行环境 120 处接收第二 UI 动作,并且该第二 UI 动作的表示可从驱动程序执行环境发送到每一个其他执行环境 130 和 140。

[0024] 可以理解,图 1 的系统 100 可实现在多个执行环境中基本上同时的 UI 测试,而不产生由于重新配置或任务切换导致的开销。因此可以理解,图 1 的系统 100 可以减少与多个执行环境所兼容的基于 UI 的应用程序的整体测试时间。

[0025] 图 2 示出了在多个执行环境中测试用户界面的系统 200 的另一个具体实施例。系统 200 包括多个计算设备(诸如,说明性计算设备 210、220 和 230)。每个计算设备可包括一个或多个执行环境。例如,第一计算设备 210 可包括第一执行环境 212,第二计算设备 220 可包括第二执行环境 222,第三计算设备 230 可包括第三执行环境 232。这些执行环境中的一个被指定为驱动程序执行环境。例如,在图 2 所示的具体实施例中,“执行环境 A”被指定为驱动程序执行环境。

[0026] 驱动程序执行环境 212 可被配置为在执行环境 212、222 和 232 的 UI 测试期间从用户 202 接收 UI 动作 204。驱动程序执行环境 212 可将 UI 动作 214 转换成 UI 动作表示 214,并将 UI 动作表示 214 发送到每一个其他执行环境 222、232。在一具体实施例中,UI 动作表示 214 由驱动程序执行环境 212 的通信代理 213 通过通信总线 240 广播到其他执行环境 222、232 的通信代理 223、233。例如,通信总线可以使用计算设备 210、220 和 230 之间的基于套接字的通信来实现。或者,通信总线 240 可以由一些其他计算设备间通信协议来实现。

[0027] 每一个其他执行环境 222 和 232 可接收 UI 动作表示 224 和 234,并基于 UI 动作表示 224 和 234 基本上同时重复 UI 动作 204。

[0028] HUD 211 可显示每一个执行环境 212、222 和 232。例如,HUD 211 可通过显示设备向用户 202 显示执行环境 212、222 和 232。在一具体实施例中,HUD 211 通过远程桌面协议(RDP)会话来显示一个或多个执行环境。例如,HUD 211 可通过分别与非驱动程序执行环境 222 和 232 的 RDP 会话 218 和 219 来显示非驱动程序执行环境 222 和 232。

[0029] HUD 211 也可从执行环境接收 UI 状态。例如,当 UI 动作 204 已经在非驱动程序执行环境 222 和 232 处重复之后,HUD 211 可从非驱动程序执行环境 222 和 232 分别接收 UI 状态 251 和 252。UI 状态 251、252 每一个可包括 UI 屏幕截图、导航状态、模型状态、自动化状态、参数状态、以及性能度量中的一个或多个。HUD 进一步参考图 5 来描述并示出。

[0030] 在一具体实施例中,第一计算设备 210 包括状态比较器 215,状态比较器 215 被配置为比较各 UI 状态。例如,状态比较器 215 可将 UI 状态 251 和 252 与驱动程序执行环境 212 的 UI 状态进行比较。状态比较器 215 也可被配置为确定 UI 状态之间何时存在状态失配。在一具体实施例中,当存在状态失配时,可以将该失配记录在日志文件 216 中。例如,可以在日志文件 216 中创建一条条目,日志文件 216 是 UI 隐错报告应用程序的一部分。HUD 211 也可向用户 202 提供状态失配的视觉指示符 206。

[0031] 在一具体实施例中,第一计算设备 210 还包括测试记录器和播放器 217。测试记录器和播放器 217 可记录多个 UI 动作,并存储多个 UI 动作表示,包括与多个 UI 动作相关联的定时信息(例如,等待时间)。所存储的表示可通过通信总线 240 被发送到执行环境。测

试记录器和播放器 217 也可被配置为基于所存储的 UI 动作的表示来重现 UI 动作。因此,当每个计算设备 210、220 和 230 均包括测试记录器和播放器时,UI 动作可以按批量方式而不是一次一个的方式来发送和重复。例如,可以在一个执行环境中记录一 UI 测试套件,并在多个执行环境中基本上同时重现该 UI 测试套件。

[0032] 在操作中,UI 动作 204(诸如,与 UI 测试相关联的 UI 动作)可在驱动程序执行环境 212 处接收,并且可在其他执行环境 222、232 处基本上同时重复。HUD 211 可显示执行环境 212、222、232,并且状态比较器 215 可确定 UI 动作 204 是否导致执行环境 212、222、232 之间的状态失配。当检测到状态失配时,HUD 211 可向用户 202 提供视觉指示符 206。

[0033] 可以理解,图 2 的系统 200 可实现在多个计算设备处(例如,在分布式计算系统处)基本上同时的 UI 测试。

[0034] 图 3 示出了数据流图 300,其示出了图 1 的系统 100 或图 2 的系统 200 处的数据流的具体实施例。

[0035] UI 测试系统处的数据流可被分为多个层。例如,用户/ HUD 监视层 310 以及总线/ 控制器层 320 可与驱动程序执行环境相关联。此外,一个或多个环境层 330 可与一个或多个非驱动程序执行环境相关联。在一说明性实施例中,用户/ HUD 监视层 310 和总线/ 控制器层 320 可与图 1 的驱动程序执行环境 120 相关联,而环境层 330 可与非驱动程序执行环境 130 和 140 中的每一个相关联。在另一个说明性实施例中,用户/ HUD 监视层 310 和总线/ 控制器层 320 可与图 2 的驱动程序执行环境 212 相关联,而环境层 330 可与图 1 的非驱动程序执行环境 222 和 232 中的每一个相关联。

[0036] 当用户 302 执行 (312) UI 动作时,在用户/ HUD 监视层 310 处开始数据流。前进到总线/ 控制器层 320, UI 动作(或其表示)可被广播 (322) 到所有非驱动程序执行环境。接着,驱动程序执行环境处的每个环境层 330 可重复 (332) 该 UI 动作(例如,以基本上同时的方式)。随后可以从每个环境层 330 向总线/ 控制器层 320 返回 (334) UI 结果。UI 结果可以在总线/ 控制器层 320 处聚集 (324),并可由用户/ HUD 监视层 310 报告 (314) 给用户 302。数据流可在层 310、320 和 330 之间以循环方式继续,直到完成所有的 UI 测试。

[0037] 图 4 示出了在多个执行环境中测试用户界面的方法 400 的具体实施例的流程图。在一说明性实施例中,方法 400 可由图 1 的系统 100 或图 2 的系统 200 来执行。

[0038] 方法 400 包括:在 402,选择与基于用户界面 (UI) 的应用程序相关联的一个或多个测试,并且在 404,选择多个执行环境。该多个执行环境中的一个被指定为驱动程序执行环境,并且至少两个执行环境在显示分辨率、语言、软件应用程序、操作系统、硬件架构、驱动程序、版本、其他环境变量、或某些其他特性方面有所不同。例如,参考图 1,可选择一或多个 UI 测试以便在执行环境 120、130 和 140 处执行。其中“执行环境 A”是驱动程序执行环境 120。

[0039] 方法 400 还包括:在 406,初始化每个执行环境处的通信代理。例如,参考图 1,通信代理 122、132 和 142 可被初始化(例如,包括通信代理的主机应用程序可在每个执行环境 120、130 和 140 处被激活)。方法 400 进一步包括:在 408,显示与驱动程序执行环境相对应的驱动程序 UI。例如,参考图 1,HUD110 可显示驱动程序执行环境 120。

[0040] 方法 400 包括:在 410,在驱动程序执行环境处接收与一个或多个测试相关联的 UI 动作。例如,参考图 1,可以在驱动程序执行环境 120 处接收 UI 动作 104。方法 400 还包括:

在 412, 将 UI 动作转换成一组输入设备控制。输入设备可以是键盘或鼠标, 而输入设备控制可以是键盘键入、鼠标移动、或鼠标点击。例如, 参考图 1, UI 动作 104 可以被转换成 UI 动作表示 124, 其中 UI 动作表示 124 包括输入设备控制。

[0041] 方法 400 进一步包括: 在 414, 将该组输入设备控制从驱动程序执行环境处的通信代理通过通信总线发送到每一个其他执行环境处的通信代理。输入设备控制可以按串行化格式来广播。UI 动作基本上同时在每一个其他执行环境处重复。例如, 参考图 1, UI 动作表示 124 可以经由通信总线 150 从通信代理 122 广播至通信代理 132、142, 并且 UI 动作 104 可以基本上同时在每一个执行环境 130、140 处重复。

[0042] 方法 400 可对与一个或多个测试相关联的每一个 UI 动作从 414 循环返回到 410。该方法在 416 结束 (例如, 当该一个或多个测试完成时)。

[0043] 图 5 是平视显示画面 (HUD) 500 的具体实施例的屏幕截图, HUD 500 显示驱动程序执行环境 502 以及其他执行环境 504、506、508 以及 510。在一个说明性实施例中, HUD 500 可包括图 1 的 HUD 110 或图 2 的 HUD 211。

[0044] 在图 5 所示出的具体实施例中, 驱动程序执行环境 502 是 64 位英语 2008 版操作系统环境, 其具有 8GB 的 RAM。第一非驱动程序执行环境 504 是 32 位英语 2010 版操作系统环境, 其具有 2GB 的 RAM。第二非驱动程序执行环境 506 是 64 位日语 2010 版操作系统环境, 其具有 4GB 的 RAM。第三非驱动程序执行环境 508 是 64 位英语 2006 版操作系统环境, 其具有 1GB 的 RAM。第四非驱动程序执行环境 510 是 32 位阿拉伯语 2008 版操作系统环境, 其具有 512MB 的 RAM。

[0045] 驱动程序执行环境 502 可在 HUD 500 处显示。用户可与驱动程序执行环境 502 进行交互 (诸如, 经由键盘、鼠标、或其他输入设备)。当用户执行某一 UI 动作时, 该 UI 动作基本上同时在其他执行环境 504、506、508 以及 510 处重复。每一其他执行环境 504、506、508 以及 510 处的 UI 状态可以被显示在 HUD 500 处。在一具体实施例中, UI 状态可以经由与其他执行环境 504、506、508 以及 510 的 RDP 会话来显示。其他执行环境 504、506、508 以及 510 可位于与驱动程序执行环境 502 相同的计算设备上, 或位于不同的计算设备上。此外, 在一具体实施例中, HUD 500 可被显示在不包括任何这些执行环境 502、504、506、508 以及 510 的计算设备上。

[0046] HUD 500 可在检测到 UI 状态失配时显示视觉指示符 512。显示视觉指示符 512 可包括: 改变与一执行环境相关联的颜色、字体或边框。例如, 在图 5 示出的具体实施例中, 在第四非驱动程序执行环境 510 处发生错误。因此, 第四非驱动程序执行环境 510 指示与驱动程序执行环境 502 的状态失配。当检测到状态失配时, HUD 500 可提供一方法以记录该状态失配。例如, HUD 500 可包括隐错提交对话框 514, 该对话框可操作来记录该状态失配。

[0047] 可以理解, 图 5 的 HUD 500 可方便地提供对被测试的每一个执行环境的同时显示。还可以理解, 图 5 的 HUD 500 可在检测到状态失配时自动提供隐错报告能力。

[0048] 图 6 描绘了包括可用于支持根据本发明的计算机实现的方法、计算机程序产品以及系统组件的各实施例的计算设备 610 的计算环境 600 的框图。在一说明性实施例中, 计算设备 610 可包括图 1 的系统 100 或其组件、图 2 的系统 200 或其组件、图 3 的层 310、320 和 330 中的一个或多个。图 1 的系统 100 或其组件、图 2 的系统 200 或其组件、以及图 3 的层 310、320 和 330 中的每一个可包括计算设备 610 或其一部分或使用计算设备 610 或其一

部分来实现。

[0049] 计算设备 610 包括至少一个处理器 620 和系统存储器 630。取决于计算设备的配置和类型,系统存储器 630 可以是易失性的(诸如随机存取存储器,即“RAM”)、非易失性的(诸如只读存储器(即“ROM”)、闪存以及即使在未提供电源时也维持已存储的数据的类似存储器设备)或两者的某种组合。系统存储器 630 通常包括:操作系统 632、一个或多个应用程序平台 634、一个或多个应用程序,以及程序数据 638。

[0050] 例如,系统存储器 630 可包括 HUD 逻辑 636 以及状态比较器 637。在一说明性实施例中,HUD 逻辑可生成并更新图 1 的 HUD 110 或图 2 的 HUD 211。HUD 可被配置为显示多个执行环境,其中一个执行环境被指定为驱动程序执行环境。HUD 还可被配置为接收驱动程序执行环境处的(例如,与 UI 测试相关联的)UI 动作,并将该 UI 动作的表示从驱动程序执行环境发送到每一个其他执行环境。UI 动作可以基本上同时在每一个其他执行环境中重复。状态比较器可比较各个执行环境的状态。

[0051] 计算设备 610 还可具有附加特征或功能。例如,计算设备 610 还可包括可移动和/或不可移动附加数据存储设备,诸如磁盘、光盘、磁带和标准大小的存储卡或闪存卡。这样的附加存储在图 6 中通过可移动存储 640 和不可移动存储 650 来示出。计算机存储介质可包括以用于存储诸如计算机可读指令、数据结构、程序组件或其它数据等信息的任何技术实现的易失性和/或非易失性存储以及可移动和/或不可移动介质。系统存储器 630、可移动存储 640 和不可移动存储 650 都是计算机存储介质的示例。计算机存储介质包括,但不限于,RAM、ROM、电可擦除可编程只读存储器(EEPROM)、闪存或其它存储器技术、紧致盘(CD)、数字多功能盘(DVD)或其它光存储、磁带盒、磁带、磁盘存储或其它磁性存储设备、或可用于存储信息且可以由计算设备 610 访问的任何其它介质。任何这样的计算机存储介质都可以是计算设备 610 的一部分。

[0052] 计算设备 610 也可包括输入设备 660,如键盘、鼠标、笔、语音输入设备、触摸输入设备等等。也可包括输出设备 670,如显示器、扬声器、打印机等等。输入设备 660 和输出设备 670 可操作来从用户 692 接收 UI 动作,并向用户 692 提供视觉指示符。计算设备 610 还包含允许计算设备 610 通过有线或无线网络与其它计算设备 690 进行通信的一个或多个通信连接 680。一个或多个通信连接 680 也可允许计算设备 610 处的多个虚拟机之间的通信。在一具体实施例中,一个或多个通信连接 680 包括图 1 的通信总线 150 或图 2 的通信总线 240。通信总线可以耦合到多个执行环境,并可在多个执行环境之间广播数据。

[0053] 可以理解,并非所有图 6 所示或以其他方式在先前的附图中描述的组件或设备都是支持此处所描述的实施例所必需的。例如,可移动存储 640 可以是可任选的。

[0054] 对此处所描述的实施方式的说明旨在提供对各实施方式的结构的大致理解。这些说明并非旨在用作对利用此处所描述的结构或方法的装置和系统的所有元件和特征的完整描述。许多其他实施例对本领域的技术人员在仔细阅读本发明后是显而易见的。可以从本发明中利用和导出其他实施方式,以使得可作出结构和逻辑替换和改变而不背离本发明的范围。因此,本发明和各附图应被认为是说明性的而非限制性的。

[0055] 本领域技术人员还可理解,结合本文所公开的各种说明性逻辑框、配置、模块、和过程步骤或指令可被实现为电子硬件或计算机软件。各种说明性组件、框、配置、模块或步骤已经大致按照其功能来描述。这种功能被实现为硬件还是软件取决于在整体系统上所施

加的具体应用和设计约束。技术人员可针对每种具体应用以不同方式来实现所描述的功能集,但此类设计决策不应被解释为致使脱离本公开的范围。

[0056] 结合此处所公开的各实施例所描述的方法或算法的各个步骤可直接用硬件、由处理器执行的软件模块、或两者的组合来实现。软件模块可以驻留在计算机可读介质中,诸如随机存储器 (RAM)、闪存、只读存储器 (ROM)、寄存器、硬盘、可移动磁盘、CD-ROM、或本领域已知的任何其他形式的存储介质。示例性存储介质耦合到处理器,使得处理器可以从该存储介质上读取信息,并向该存储介质写入信息。在替换方案中,存储介质可以集成到处理器或者处理器和存储介质可以作为分立组件驻留在计算设备或计算机系统中。

[0057] 尽管已经在此示出和描述了具体实施方式,但应理解,可以为所示的具体实施方式替换被设计成达到相同或相似目的的任何后续安排。本发明旨在覆盖各实施方式的任何和所有后续改变和变体。

[0058] 提交本发明的摘要的同时要明白,将不用它来解释或限制权利要求的范围或含义。另外,在前面的具体实施方式中,可以出于将本发明连成一个整体的目的而将各种特征组合或描述在一起放在单个实施方式中。此发明将不被解释为反映所要求保护的实施方式要求比每个权利要求中明确陈述的更多特征的意图。相反,如以下权利要求反映的,本发明的主题可涉及少于所公开的实施方式中的任一个的所有特征。

[0059] 提供前面对各实施方式的描述是为了使本领域技术人员能制作或使用各实施方式。对这些实施例的各种修改对于本领域技术人员将是显而易见的,并且本文中定义的普适原理可被应用于其他实施例而不会脱离本公开的范围。因此,本发明不是旨在限于本文所示的各实施方式,而是按照与如由以下权利要求书定义的原理和新颖特征相一致的尽可能最宽范围。

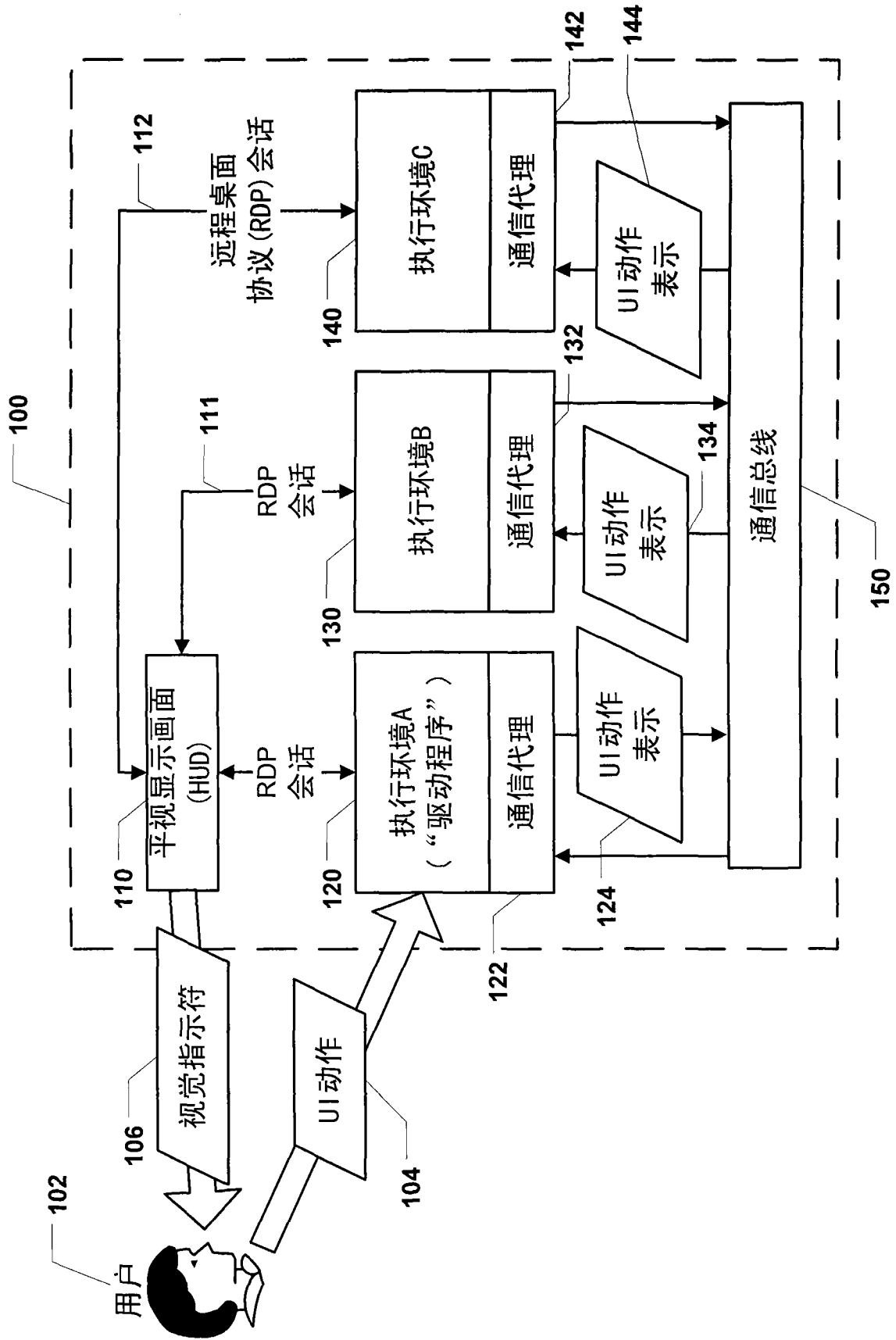


图 1

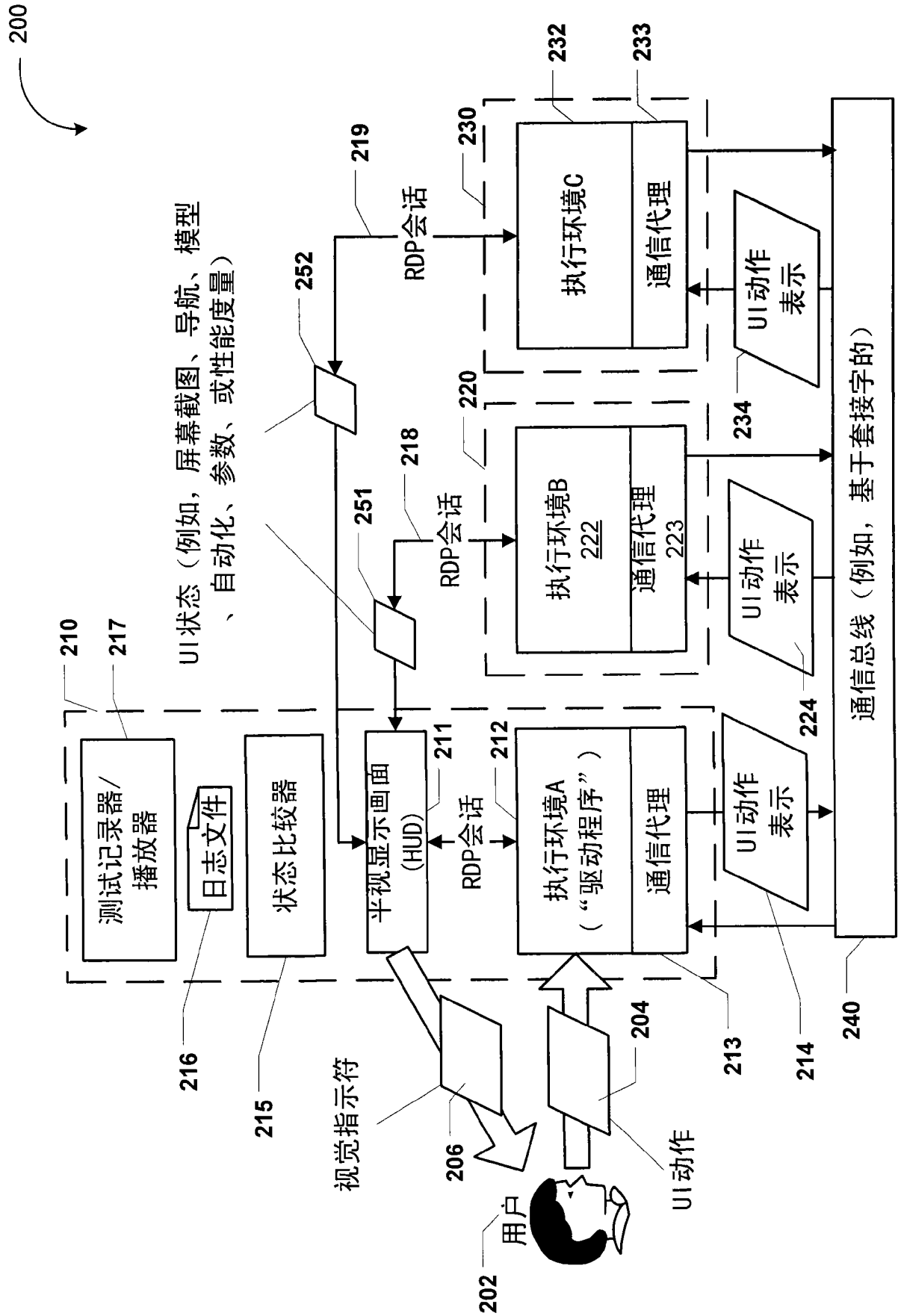


图 2

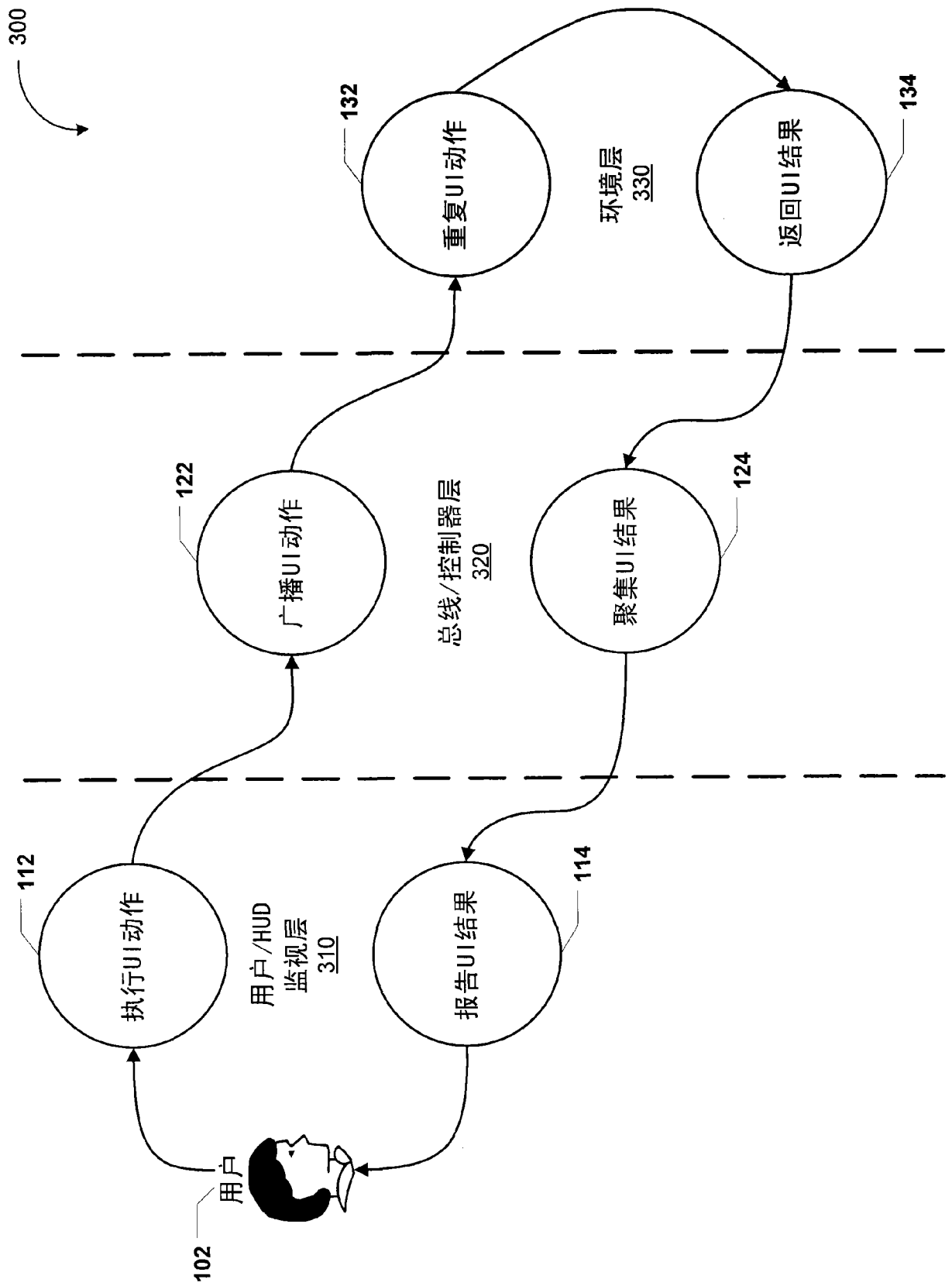


图 3

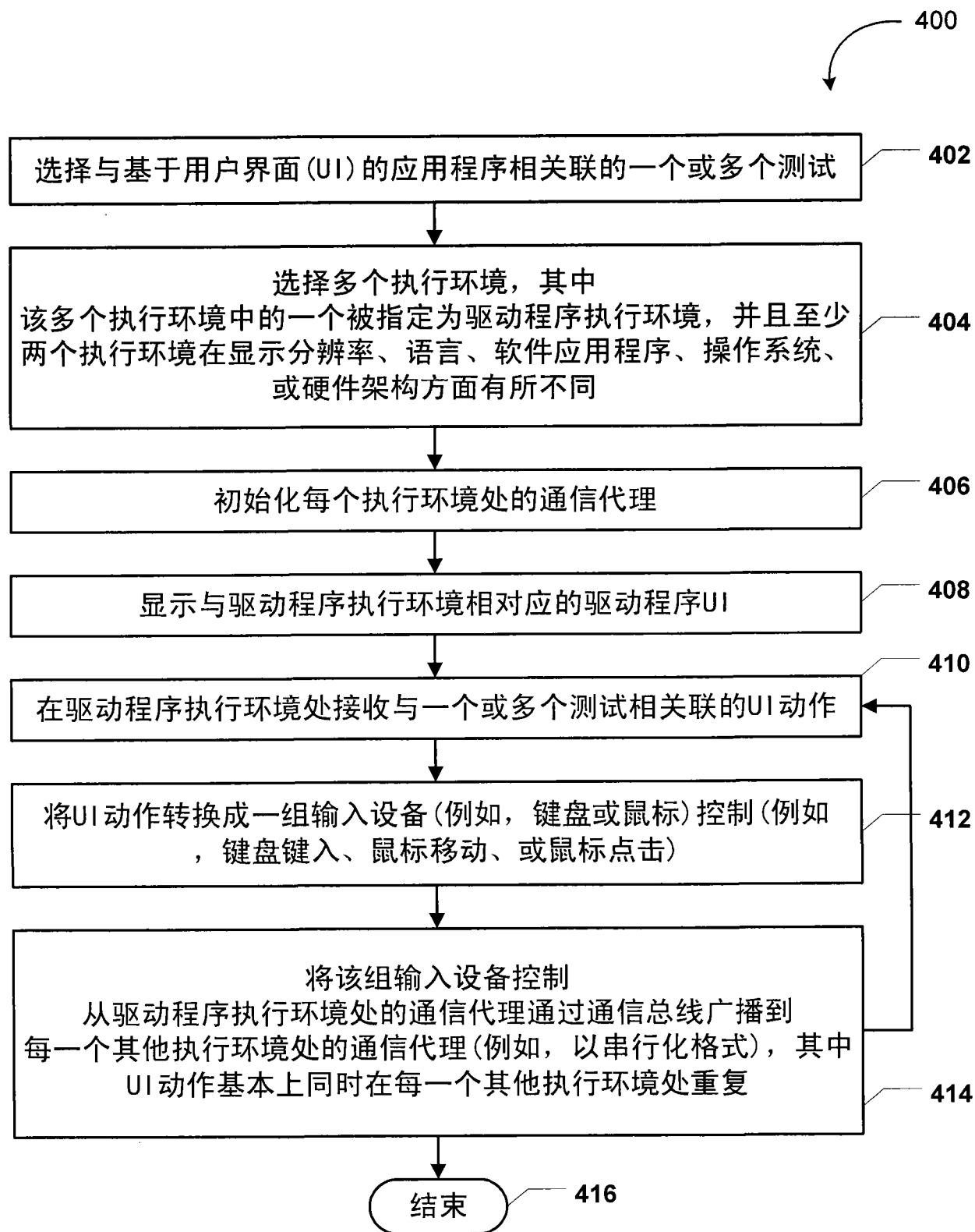


图 4

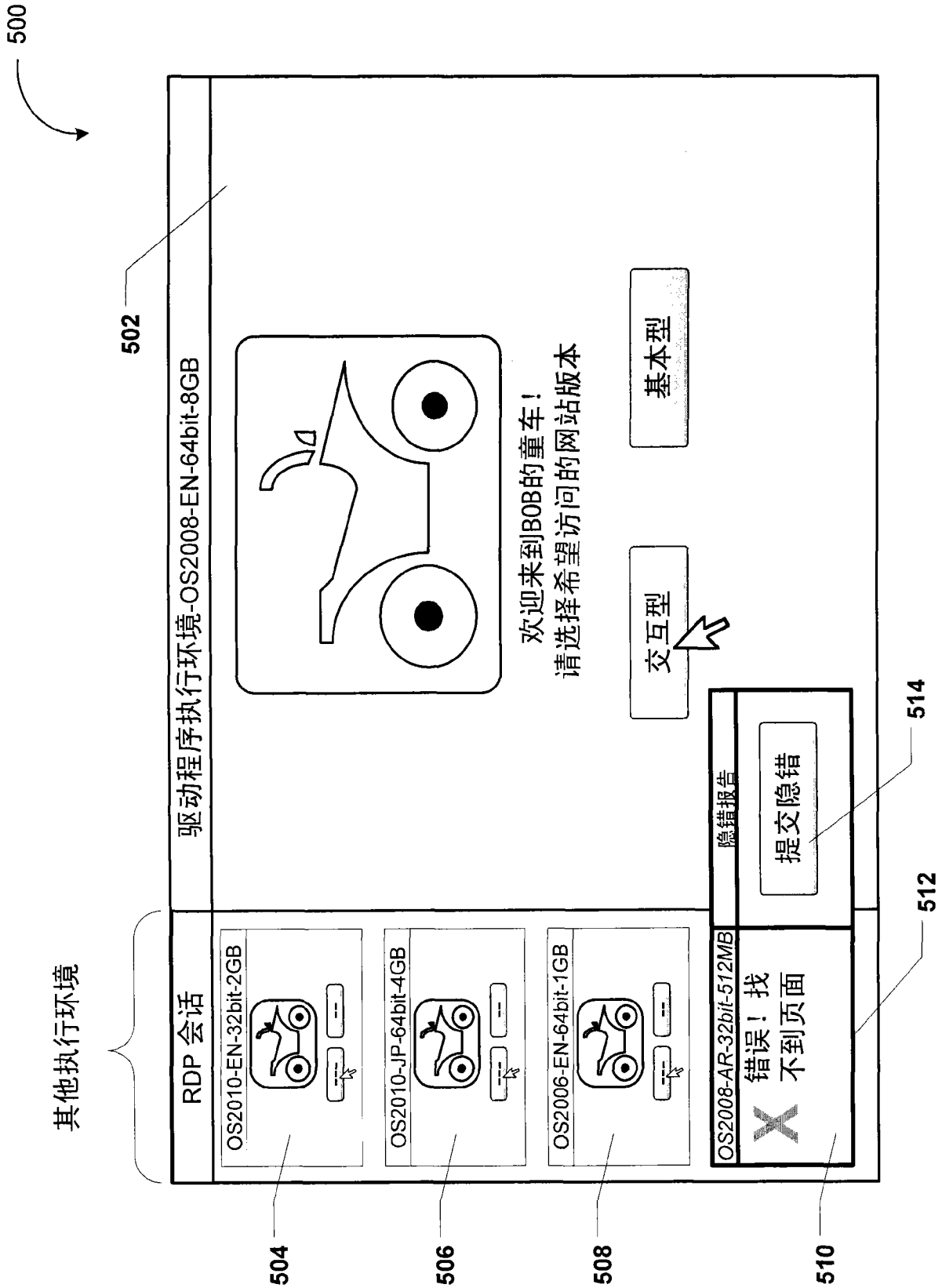


图 5

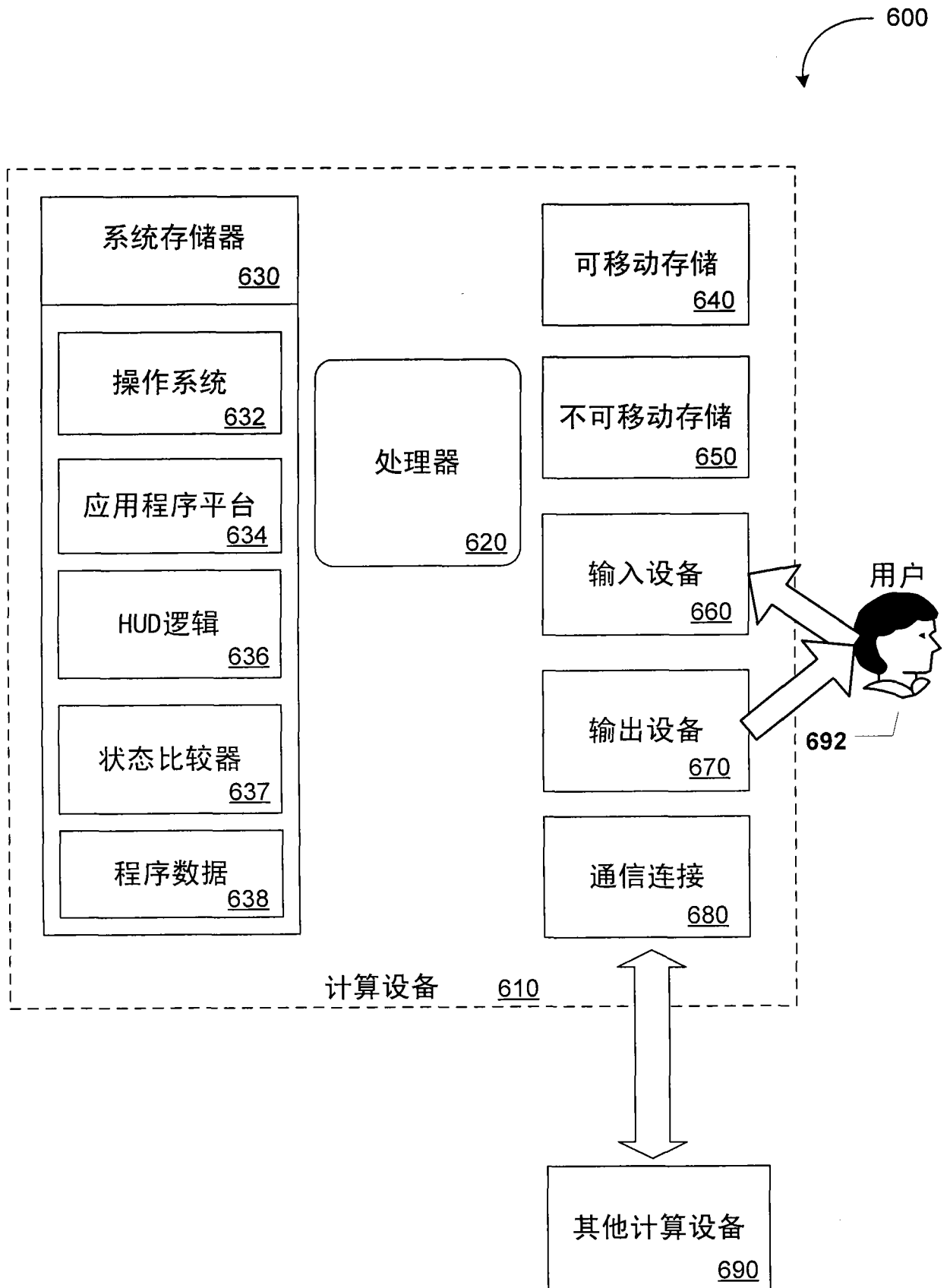


图 6