(19)国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 115943368 A
(43)申请公布日 2023.04.07

(21)申请号 202180004289.2

(22)申请日 2021.10.29

(30)优先权数据
17/340,603 2021.06.07 US
17/341,550 2021.06.08 US

(85)PCT国际申请进入国家阶段日
2021.12.29

(86)PCT国际申请的申请数据
PCT/US2021/057166 2021.10.29

(87)PCT国际申请的公布数据
WO2022/260696 EN 2022.12.15

(71)申请人 尤帕斯公司
地址 美国纽约州

(72)发明人 M·格里戈尔

(74)专利代理机构 北京市金杜律师事务所 11256
专利代理师 罗利娜

(51)Int.Cl.
G06F 9/455(2006.01)
G06F 9/448(2006.01)
G06F 8/34(2006.01)
G06Q 10/10(2006.01)

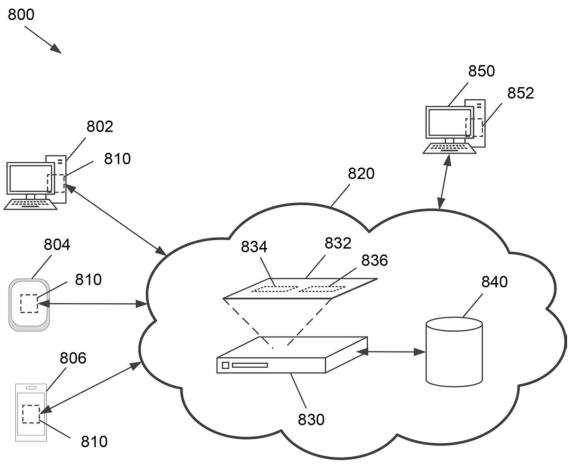
权利要求书3页 说明书18页 附图18页

(54)发明名称

基于Web的机器人流程自动化设计器系统以及用于虚拟机、会话和容器的自动化

(57)摘要

公开了基于web的机器人流程自动化(RPA)设计器系统，该系统允许RPA开发者设计并实现web无服务器自动化、用户界面(UI)自动化和其他自动化。这种基于web的RPA设计器系统可以允许开发者通过云登录并获得模板项目、开发者设计项目、服务、活动等的列表。因此，RPA开发可以是集中式的并基于云的，这减少了用户计算系统上的本地处理和存储器需求，并且集中了RPA设计器功能，能够实现更好的合规性。由基于web的RPA设计器系统生成的自动化可以在虚拟机(VM)、容器或操作系统会话中被部署并被执行。



CN 115943368 A

1.一种基于云的系统,包括:

存储器,存储计算机程序指令;以及

至少一个处理器,被配置为执行所述计算机程序指令,其中所述计算机程序指令被配置为使得所述至少一个处理器:

在运行时在第一web浏览器中运行自动化,所述第一web浏览器位于所述基于云的系统的操作系统会话、虚拟机(VM)或容器中。

2.根据权利要求1所述的基于云的系统,其中所述自动化是无服务器自动化。

3.根据权利要求1所述的基于云的系统,其中所述自动化作为所述第一web浏览器的标签的一部分被执行。

4.根据权利要求1所述的系统,其中所述自动化是在没有RPA机器人的情况下被执行的。

5.根据权利要求1所述的基于云的系统,其中

所述自动化被配置为使得所述第一web浏览器向客户端计算系统的第二web浏览器的web扩展发送一个或多个命令,并且

所述web扩展被配置为:

与所述客户端计算系统的所述第二web浏览器交互,从所述客户端计算系统的所述第二web浏览器的一个或多个标签中的一个或多个网页获得信息,或者两者兼有;以及

向所述第一web浏览器提供对所请求的交互已发生的确认,提供所获得的信息,或者两者兼有。

6.根据权利要求5所述的基于云的系统,其中,所述web扩展被配置为提供本地功能,所述本地功能在所述自动化经由所述第一web浏览器在所述基于云的系统的所述容器中运行时,被无头地执行。

7.根据权利要求5所述的基于云的系统,其中,所述第一web浏览器被配置为使用对与所述web浏览器的所述交互的所述确认、所述所获得的信息、或所述确认和所述所获得的信息两者,来执行与所述自动化相关联的流程。

8.根据权利要求1所述的基于云的系统,其中

所述自动化被配置为使得所述第一web浏览器向客户端计算系统的本地RPA扩展流程发送一个或多个请求,并且

所述本地RPA扩展流程被配置为:

与所述客户端计算系统的一个或多个应用和/或流程交互,从运行在所述客户端计算系统上的所述一个或多个应用和/或流程获得信息,或者两者兼有;以及

向所述第一web浏览器提供对所请求的交互已发生的确认,提供所获得的信息,或者两者兼有。

9.根据权利要求8所述的基于云的系统,其中,所述第一web浏览器被配置为使用对所述所请求的交互已发生的所述确认、所述所获得的信息、或所述确认和所述所获得的信息两者,来执行与所述自动化相关联的流程。

10.根据权利要求1所述的基于云的系统,其中,所述计算机程序指令还被配置为使得所述至少一个处理器:

检测用于执行所述自动化的触发事件、条件或命令;

由所述第一web浏览器向第二web浏览器的web扩展发送一个或多个命令,以与所述第二web浏览器交互,从所述第二web浏览器的一个或多个标签中的一个或多个网页获得信息,或者两者兼有;

由所述第一web浏览器从所述web扩展接收与所述第二web浏览器的所述交互的结果、所获得的信息、或所述结果和所述所获得的信息两者;以及

由所述第一web浏览器使用接收的与所述第二web浏览器交互的所述结果、所述所获得的信息或所述结果和所述所获得的信息两者,来执行与所述自动化相关联的流程。

11.根据权利要求1所述的基于云的系统,其中,计算机程序指令还被配置为使得所述至少一个处理器:

检测用于执行所述自动化的触发事件、条件或命令;

由所述第一web浏览器向客户端计算系统的本地RPA扩展流程发送一个或多个请求,以与所述客户端计算系统的一个或多个应用和/或流程进行交互,从运行在所述客户端计算系统上的所述一个或多个应用和/或流程获得信息,或者两者兼有;

由所述第一web浏览器接收对所请求的交互已发生的确认,接收所获得的信息,或者两者兼有;以及

由所述第一web浏览器使用接收的对所述所请求的交互已发生的所述确认、所述所获得的信息、或所述确认和所述所获得的信息两者,来执行与所述自动化相关联的流程。

12.一种非瞬态算机可读介质,存储计算机程序,所述计算机程序被配置为使得至少一个处理器:

在运行时在第一web浏览器中运行自动化,所述第一web浏览器位于基于云的系统的操作系统会话、虚拟机(VM)或容器中,其中

所述自动化被配置为使得RPA机器人向客户端计算系统的第二web浏览器的web扩展发送一个或多个命令,并且

所述web扩展被配置为:

与所述客户端计算系统的所述第二web浏览器交互,从所述客户端计算系统的所述第二web浏览器的一个或多个标签中的一个或多个网页获得信息,或者两者兼有;以及

向所述第一web浏览器提供对所请求的交互已发生的确认,提供所获得的信息,或者两者兼有。

13.根据权利要求12所述的非瞬态计算机可读介质,其中所述自动化是无服务器自动化。

14.根据权利要求12所述的非瞬态计算机可读介质,其中所述自动化作为所述第一web浏览器的标签的一部分被执行。

15.根据权利要求12所述的非瞬态计算机可读介质,其中所述自动化在没有RPA机器人的情况下被执行。

16.根据权利要求12所述的非瞬态计算机可读介质,其中所述web扩展被配置为提供本地功能,所述本地功能在所述自动化经由所述第一web浏览器在所述基于云的系统的所述容器中运行时被无头地执行。

17.根据权利要求12所述的非瞬态计算机可读介质,其中所述第一web浏览器被配置为使用对与所述web浏览器的所述交互的所述确认、所获得的信息、或所述确认和所述所获得

的信息两者，来执行与所述自动化相关联的流程。

18.根据权利要求12所述的非瞬态计算机可读介质，其中，所述计算机程序还被配置为使得所述至少一个处理器：

检测用于执行所述自动化的触发事件、条件或命令；以及

响应于检测到的所述触发事件、条件或命令，在所述第一web浏览器中运行所述自动化。

19.一种计算机实现的方法，包括：

由计算系统在运行时在web浏览器中运行自动化，所述web浏览器位于基于云的系统的操作系统会话、虚拟机(VM)或容器中，其中

所述自动化被配置为使得所述web浏览器向客户端计算系统的本地RPA扩展流程发送一个或多个请求，并且

所述本地RPA扩展流程被配置为：

与所述客户端计算系统的一个或多个应用和/或流程交互，从运行在所述客户端计算系统上的所述一个或多个应用和/或流程获得信息，或者两者兼有；以及

向所述web浏览器提供对所请求的交互已发生的确认，提供所获得的信息，或者两者兼有。

20.根据权利要求19所述的计算机实现的方法，其中所述web浏览器被配置为使用对所述所请求的交互已发生的所述确认、所述所获得的信息、或所述确认和所述所获得的信息两者，来执行与所述自动化相关联的流程。

21.根据权利要求19所述的计算机实现方法，其中所述自动化是无服务器自动化。

22.根据权利要求19所述的计算机实现的方法，其中所述自动化被作为所述web浏览器的标签的一部分被执行。

23.根据权利要求19所述的计算机实现的方法，其中所述自动化在没有RPA机器人的情况下被执行。

24.根据权利要求19所述的计算机实现的方法，还包括：

由所述计算系统检测用于执行所述自动化的触发事件、条件或命令；以及

由所述计算系统响应于检测到的所述触发事件、条件或命令，在所述web浏览器中运行所述自动化。

基于Web的机器人流程自动化设计器系统以及用于虚拟机、会话和容器的自动化

[0001] 相关申请的交叉引用

[0002] 本申请要求于2021年6月8日提交的美国专利申请No.17/341550的优先权，美国专利申请No.17/341550要求于2021年6月7日提交的美国专利申请No.17/340603的优先权，并且是美国专利申请No.17/340603的延续申请。这些早些时候提交的申请的主题在此全文引入作为参考。

技术领域

[0003] 本发明总体上涉及机器人流程自动化(RPA)，更具体地，涉及基于web的RPA设计器系统，其允许RPA开发者设计并实现web无服务器自动化、用户界面(UI)自动化以及其他自动化。

背景技术

[0004] 许多组织的政策不允许员工直接在他们自己的计算系统上安装附加软件或升级现有软件。这种政策可以适当地尝试确保遵守美国和其他国家的法律或协议，诸如欧盟通用数据保护条例(GDPR)、美国健康保险可移植性和责任法案(HIPAA)、第三方服务条款等。当前的RPA开发软件作为本地驻留在用户的计算系统上的桌面工具而存在，在向用户推出新版本或增强之前，这可能需要软件的新版本或新能力由IT进行测试并由合规官进行审查。此外，许多新能力和服务都是基于云的。相应地，改进的和/或替代的RPA方法可能是有益的。

发明内容

[0005] 本发明的某些实施例可以为本领域中尚未被当前的RPA技术完全标识、理解或解决的问题和需求提供解决方案。例如，本发明的一些实施例涉及基于web的RPA设计器系统，该基于web的RPA设计器系统允许RPA开发者设计并实现web无服务器自动化、用户界面(UI)自动化和其他自动化web。

[0006] 在一个实施例中，一种系统包括：开发者计算系统，包括web浏览器；以及开发服务器，提供基于web的RPA设计器应用。web浏览器被配置为访问并显示基于web的RPA设计器应用的web界面。web界面被配置为创建RPA项目，配置RPA工作流程，以及向基于web的RPA设计器应用提交RPA工作流程配置。基于web的RPA设计器应用被配置为向web浏览器提供web界面，基于从web浏览器提交的RPA工作流程配置生成自动化，经由RPA机器人执行所生成的自动化并验证所生成的自动化，以及向web浏览器提供由RPA机器人执行的结果和所成的自动化的验证的结果。所生成的自动化被配置为由RPA机器人在操作系统会话、VM或容器中远程地执行。

[0007] 在另一实施例中，一种非瞬态计算机可读介质存储用于基于web的RPA设计器应用的计算机程序。该计算机程序被配置为使得至少一个处理器向计算系统的web浏览器提供

web界面。web界面被配置为提供用以创建RPA项目、配置RPA工作流程、以及向基于web的RPA设计器应用提交RPA工作流程配置的功能。该计算机程序还被配置为使得至少一个处理器基于从web浏览器提交的RPA工作流程配置生成自动化，经由RPA机器人执行所生成的自动化并验证所生成的自动化。该计算机程序还被配置为使得至少一个处理器向web浏览器提供由RPA机器人执行的结果和所生成的自动化的验证的结果。所生成的自动化被配置为由生产RPA机器人在运行时在操作系统会话、VM或容器中远程地执行。

[0008] 在又一实施例中，一种用于基于web的RPA的计算机实现方法包括：由基于云的计算系统基于从web浏览器提交的RPA工作流程配置生成自动化，由基于云的计算系统经由RPA机器人执行所生成的自动化并验证所生成的自动化。该计算机实现的法还包括由基于云的计算系统向web浏览器提供由RPA机器人执行的结果和所生成的自动化的验证的结果。所生成的自动化被配置为由生产RPA机器人在运行时在操作系统会话、VM或容器中远程地执行。

[0009] 在另一实施例中，一种基于云的系统包括：存储器，存储计算机程序指令；以及至少一个处理器，被配置为执行计算机程序指令。该计算机程序指令被配置为使得至少一个处理器运行多个RPA机器人作为运行时服务。多个RPA机器人位于基于云的系统的操作系统会话、VM或容器中。该计算机程序指令还被配置为使得至少一个处理器由多个RPA机器人执行自动化。多个自动化可由多个RPA机器人访问，并且位于基于云的生产服务器的操作系统会话、VM或容器中，或者位于基于云的系统的远程。

[0010] 在又一实施例中，一种非瞬态计算机可读介质存储计算机程序。该计算机程序被配置为使得至少一个处理器运行RPA机器人作为运行时服务。RPA机器人位于操作系统会话、VM或容器中。该计算机程序还被配置为使得至少一个处理器经由RPA机器人执行自动化。自动化被配置为使得RPA机器人向客户端计算系统的web浏览器的web扩展发送一个或多个命令。web扩展被配置为与客户端计算系统的web浏览器交互，从客户端计算系统的web浏览器的一个或多个标签中的一个或多个网页获得信息，或者两者兼有。Web扩展还被配置为向RPA机器人提供对所请求的交互已发生的确认，提供所获得的信息，或者提供对所请求的交互已发生的确认和所获得的信息两者。

[0011] 在又一实施例中，一种计算机实现方法包括由基于云的计算系统运行RPA机器人作为运行时服务。RPA机器人位于计算系统的操作系统会话、VM或容器中。该计算机实现的方法还包括由RPA机器人执行自动化。自动化被配置为使得RPA机器人向客户端计算系统的本地RPA扩展流程发送一个或多个请求。本地RPA扩展流程被配置为与客户端计算系统的一个或多个应用和/或流程交互，从运行在客户端计算系统上的一个或多个应用和/或流程获得信息，或者两者兼有。本地RPA扩展流程还被配置为向RPA机器人提供对所请求的交互已发生的确认，提供所获得的信息，或者提供对所请求的交互已发生的确认和所获得的信息两者。

[0012] 在另一实施例中，一种系统包括：开发者计算系统，包括web浏览器；以及开发服务器，提供基于web的RPA设计器应用。该基于web的RPA设计器应用被配置为使得web浏览器下载用于基于web的RPA设计器应用的代码，并且在web浏览器中显示用于基于web的RPA设计器应用的web界面。web界面被配置为提供用以创建RPA项目和配置RPA工作流程的功能。该基于web的RPA设计器应用还被配置为使得web浏览器在web浏览器中为经配置的RPA工作流

程生成自动化，并且在web浏览器中执行并验证所生成的自动化。

[0013] 在又一实施例中，一种非瞬态计算机可读介质存储计算机程序。该计算机程序被配置为使得至少一个处理器将用于基于web的RPA设计器应用的代码下载到web浏览器，并且在web浏览器中显示用于基于web的RPA设计器应用的web界面。Web界面被配置为提供用以创建RPA项目和配置RPA工作流程的功能。该计算机程序还被配置为使得至少一个处理器在web浏览器中为经配置的RPA工作流程生成自动化，并且在web浏览器中执行并验证所生成的自动化。

[0014] 在又一实施例中，一种计算机实现方法包括由计算系统将用于基于web的RPA设计器应用的代码下载到web浏览器。该计算机实现方法还包括由计算系统在web浏览器中显示用于基于web的RPA设计器应用的web界面。Web界面被配置为提供用以创建RPA项目和配置RPA工作流程的功能。该计算机实现方法还包括由计算系统在web浏览器中为经配置的RPA工作流程生成自动化，以及由计算系统在web浏览器中执行并验证所生成的自动化。用于执行并验证所生成的自动化的代码被包括在下载的代码中。

[0015] 在另一实施例中，一种基于云的系统包括：存储器，存储计算机程序指令；以及至少一个处理器，被配置为执行计算机程序指令。该计算机程序指令被配置为使得至少一个处理器在运行时在第一web浏览器中运行自动化。第一web浏览器位于基于云的系统的操作系统会话、VM或容器中。

[0016] 在又一实施例中，一种非瞬态计算机可读介质存储计算机程序。该计算机程序被配置为使得至少一个处理器在运行时在第一web浏览器中运行自动化。第一web浏览器位于基于云的系统的操作系统会话、VM或容器中。自动化被配置为使得RPA机器人向客户端计算系统的第二web浏览器的web扩展发送一个或多个命令。Web扩展被配置为与客户端计算系统的第二web浏览器交互，从客户端计算系统的第二web浏览器的一个或多个标签中的一个或多个网页获得信息，或者两者兼有。Web扩展还被配置为向第一web浏览器提供对所请求的交互已发生的确认，提供所获得的信息，或者提供对所请求的交互已发生的确认和所获得的信息两者。

[0017] 在又一实施例中，一种计算机实现方法包括由计算系统在运行时在web浏览器中运行自动化。该web浏览器位于基于云的系统的操作系统会话、VM或容器中。自动化被配置为使得web浏览器向客户端计算系统的本地RPA扩展流程发送一个或多个请求。本地RPA扩展流程被配置为与客户端计算系统的一个或多个应用和/或流程交互，从运行在客户端计算系统上的一个或多个应用和/或流程获得信息，或者两者兼有。本地RPA扩展流程还被配置为向web浏览器提供对所请求的交互已发生的确认，提供所获得的信息，或者提供对所请求的交互已发生的确认和所获得的信息两者。

附图说明

[0018] 为了容易理解本发明的某些实施例的优点，将通过参考附图中示出的具体实施例来呈现上面简要描述的本发明的更具体的描述。虽然应当理解，这些附图仅描绘了本发明的典型实施例，并且因此不应被认为是对其范围的限制，但是将通过使用附图以附加的特征和细节来描述并解释本发明，其中：

[0019] 图1是示出根据本发明实施例的基于web的RPA设计器系统的架构图。

[0020] 图2是示出了根据本发明实施例的计算系统的架构图，该计算系统被配置为实现基于web的RPA设计器系统的部分或全部。

[0021] 图3A是示出根据本发明实施例的基于web的RPA系统的架构图。

[0022] 图3B示出了根据本发明实施例的在容器、VM或操作系统会话中运行的自动化和RPA机器人。

[0023] 图4A示出了根据本发明实施例的基于web的RPA开发应用网页的主页视图。

[0024] 图4B示出了根据本发明实施例的显示有新自动化窗口的基于web的RPA开发应用网页的主页视图。

[0025] 图4C示出了根据本发明实施例的基于web的RPA开发应用网页的画布视图。

[0026] 图4D示出了根据本发明实施例的具有活动类别和快捷动作窗口的基于web的RPA开发应用网页的画布视图。

[0027] 图4E示出了根据本发明实施例的具有Excel®活动窗口的基于web的RPA开发应用网页的画布视图。

[0028] 图4F示出了根据本发明实施例的具有未配置的使用Excel®文件活动的基于web的RPA开发应用网页的画布视图。

[0029] 图4G示出了根据本发明的实施例的在已经配置使用Excel®文件活动之后的基于web的RPA开发应用网页的画布视图。

[0030] 图5是示出根据本发明实施例的用于在设计时执行基于web的RPA开发的流程的流程图。

[0031] 图6是示出根据本发明实施例的在运行时执行与web扩展交互的自动化的流程的流程图。

[0032] 图7是示出根据本发明实施例的在运行时执行在VM或容器中运行的自动化的流程的流程图。

[0033] 图8是示出根据本发明实施例的被配置为与本地应用交互的基于web的RPA系统的架构图。

[0034] 图9是示出根据本发明实施例的在运行时执行与本地RPA扩展流程交互的自动化的流程的流程图，该本地RPA扩展流程运行在不同于RPA机器人的计算系统上。

[0035] 图10是示出根据本发明实施例的在设计时经由web浏览器执行基于web的RPA开发而无需RPA机器人的流程的流程图。

[0036] 图11是示出根据本发明实施例的在运行时在容器、VM或操作系统会话的web浏览器中执行自动化的流程的流程图。

[0037] 除非另有说明，否则在所有附图中，相似的附图标记始终表示相应的特征。

具体实施方式

[0038] 一些实施例涉及基于web的RPA设计器系统，该基于web的RPA设计器系统允许RPA开发者设计并实现web无服务器自动化、UI自动化和其他自动化。在一些实施例中，例如，在基于云的环境中提供基于web的RPA设计器系统、由该系统生成的自动化、或基于web的RPA设计器系统和由该系统生成的自动化两者。这种实施例可以不要求桌面安装，并且可以允

许用户通过云登录并获得模板项目、开发者设计项目、服务、活动等的列表。因此，在一些实施例中，RPA开发可以是集中的并基于云的，这减少了用户计算系统上的本地处理和存储器需求，并且集中了RPA设计器功能，实现了更好的合规性。这种实施例还可以使RPA服务提供商能够在其端维护和升级RPA设计器系统，这减少了维护问题以及与升级相关联的成本，并且由客户增加新功能。这还能够实现更快且更规则的更新、新特征的推出、漏洞修复等。

[0039] 在一些实施例中，RPA开发者可以经由web浏览器访问基于云的RPA设计器系统。他或她最初可以会看到主页(例如，参见图7A)，从该主页中可以创建项目并且可以提供关于用户想要构建什么的信息。然后可以会出现“画布”网页(例如，参见图7C)，在该“画布”网页上，用户可以构建期望的RPA自动化。例如，在一些实施例中，用户可以配置RPA工作流程中的RPA活动(activity)，并且控制自动化将如何被运行(例如，手动地、基于触发、调度等)。在按预期设计RPA工作流程之后，在一些实施例中，用户可以发布RPA工作流程，并且可以按照预期部署自动化。

[0040] 在一些实施例中，基于web的RPA设计器系统为用户提供以下能力：设计、编辑和测试服务器侧工作流程、无服务器工作流程、长时间运行工作流程(例如，可以等待某些流程完成或者恢复操作之前的某些输入的工作流程)、编排流程、无头(headless)工作流程、后台(background)流程、非UI自动化工作流程、通常不要求用户交互的工作流程等。例如，在基于web的RPA设计器系统中创建的无服务器工作流程可以被保存到云，并且用户可以经由web浏览器手动地运行和调试服务器侧工作流程。这些包可以被发布到云(例如，服务器侧)编排器(orchestrator)和助手。在作为包被发布时，项目可以被编译。

[0041] 一些实施例中的触发是应用编程接口(API)触发。API触发是程序到程序的调用，其可能来自应用或操作系统(OS)，也可以来自各种用户或软件事件，诸如用户与用户界面交互(例如，点击按钮)或输入某些信息、对已经接收到电子邮件的通知、屏幕上出现的图元素、按键被按下、文件改变、元素属性改变、OS级别事件等。触发可以被实现为类，其中事件是类的实例，并且可以经由事件监听器来检测事件。

[0042] 一些实施例中的基于web的RPA设计器系统可以开发UI自动化，UI自动化被配置为响应至少两种类型的API触发(诸如推送触发和拉取触发)。拉取触发可以重复运行(例如，每次OS在寻找信息的流程中循环时、每秒钟、每分钟、每十分钟等)并且检查是否发生了与该触发相关的改变(诸如指示已接收到新电子邮件的标志)、检查文本字段的值以查看文本是否改变等。推送触发是应用生成事件(例如，web钩子(hook))，当它们被触发时，使得监听器做出反应。这也可以是响应于接收到新电子邮件、由于来自服务器的请求、由于用户点击按钮等而发生的。

[0043] Web触发与“桌面”(本地)触发之间也有区别。桌面触发往往基于用户的计算系统上正在发生什么。另一方面，web触发发生在web服务器上并且可以是全局的。多个客户端(例如，两个、十个、所有等)可以接收web触发。

[0044] 然而，在一些实施例中，UI自动化可以从存在于web上的应用之上的web界面进行构建。例如，如果在Workday®的网页上有标签(tab)，则RPA开发者可以使用一些实施例中的基于云的RPA设计器系统来与该页面交互并在其上创建自动化。这种自动化可以在服务器侧(例如，在云中)运行并在虚拟机(VM)中执行，如本文后面更详细讨论的。多个VM可以在单个服务器上运行，并且每个VM通常都有其自己的操作系统和应用。

[0045] 在一些实施例中，针对UI自动化，可以提供允许用户与按钮、文本字段等交互的UI元素的分类法(taxonomy)。分类法可以是分层的，并且指定UI元素之间的关系，诸如哪些元素位于哪个页面和/或页面的哪个部分中。例如，用户可以根据分类法选择“OK”按钮，并且使自动化点击该按钮。

[0046] 在一些实施例中，基于web的UI自动化由web浏览器扩展(例如，Google Chrome®扩展、另一web浏览器的插件、或者用于扩展web浏览器的功能的任何其他合适的机制)来辅助，web浏览器扩展可以被安装并与web浏览器中的标签通信，以获得信息并在本地实现UI自动化。例如，这种扩展可以能够点击浏览器，从浏览器元素中获取文本等，并将这些信息提供给RPA机器人。在一些实施例中，web浏览器和RPA设计器应用都可以位于RPA开发者的计算系统上。然而，在某些实施例中，基于web的RPA设计器应用位于远程，并且RPA开发者经由web浏览器与基于web的RPA设计器应用交互。这可以使基于web的RPA设计器应用能够由创建RPA设计器应用的公司维护和升级，并自动地使应用保持最新。

[0047] 在运行时，扩展可以位于用户的计算系统上，并且自动化代码可以位于远程(例如，在基于云的环境中的服务器上等)。操作系统会话可以存在于用户的本地计算系统上，在该计算系统中web应用是打开的并且正在运行。UI自动化经由web扩展指示web应用执行期望的动作。如上所述，在一些实施例中，自动化可以在服务器上的VM中运行，并且不需要与用户计算系统上的web浏览器扩展交互来实现其功能。

[0048] 在一些实施例中，可以使用“无头浏览器”，其中不要求在Windows®或一些其他操作系统上的交互会话。在一些实施例中，这种无头浏览器可以在任何OS上(诸如在Linux容器中)运行。容器包含应用，使得应用与它们运行于其上的托管系统是隔离的。自动化以及它使用的库和其他依赖关系可以被打包到容器中。自动化可以用基于web的RPA设计器系统来设计，但是然后可以从API调用，并且调用可以被转换成无服务器会话。因此，无头浏览器自动化不要求例如开放的web浏览器、交互式会话和Windows®VM。

[0049] 在一些实施例中，例如，无服务器会话可以在容器、VM或Windows®会话中。计算系统可以用其硬件资源处理尽可能多的无服务器会话。例如，这些无服务器工作流程可以在操作系统中表现为正在运行的流程。

[0050] 为了设计无头浏览器的UI自动化，可以提供类似于桌面RPA设计器应用的设计体验，其中，用户与web浏览器的交互可以被记录并且/或者指示屏幕上的功能、捕获元素功能等。记录器可以位于用户正在与之交互的当前浏览器标签之上的一级。这可以经由Chrome®扩展或一些其他浏览器扩展来实现，这些其他浏览器扩展提供对浏览器本身的访问。该扩展可以与打开的标签交互，以在设计时为加载的页面设计UI自动化(例如，点击元素、创建表格等)。例如，一旦被发布或以其他方式可供运行时使用，无头浏览器UI自动化就可以被调度以在Linux®容器中被执行，而不是在VM会话中被执行。虽然考虑到它是如何设计的在这里将其称为“UI自动化”，但是当无头浏览器UI自动化运行的时候，实际上不会向用户显示任何UI。

[0051] 一些实施例中的基于web的RPA设计器系统允许用户构建UI自动化和其他自动化。如本文所使用的，“自动化”是实现RPA工作流程逻辑的代码，诸如由一些实施例中的基于

web的RPA设计器系统开发的那些。其中运行自动化的环境可以取决于自动化做什么。如果自动化是旨在执行UI交互时可见和/或要求用户进行输入的、有人值守自动化，则可以使用安装有Windows®的VM，以便UI自动化可以成功运行。然而，如果UI自动化不需要可见或不要求用户进行输入，诸如无头浏览器自动化，则这种自动化可以在例如Linux®容器中运行。Web扩展可以被用于基于web的应用，在基于web的应用中，需要来自终端用户的计算系统的信息，并且UI自动化是无头的。否则，如果需要，则可以通过启动VM来实现该功能。

[0052] 在一些实施例中，当RPA开发者想要创建UI自动化或另一自动化并直接运行它而不在用户的计算系统上本地安装和运行自动化时，使用VM或Linux®容器。将以其他方式发生在用户的计算系统上的动作转而发生在VM或容器中。为了管理这种VM或容器，可以使用自动或手动预置的池。自动预置的池可以是弹性的，并且池中的VM或容器的数量可以根据需求而增加或减少。另一方面，手动预置的池可以包括人工(例如，管理员)指定的VM或容器的数量。

[0053] “无服务器”(无人值守)自动化不需要典型的服务器来运行，并且例如可以在Linux®容器中被运行。使用一些实施例中的基于web的RPA设计器系统开发的无服务器自动化可以被调度，以在特定时间被运行，或者以其他方式在没有用户交互的情况下被运行。例如，当接收到某个电子邮件时，可以经由Slack®发送消息。当作业在指挥器应用上失败时，可以重新运行该作业，或者如果该作业因特定错误而失败，则可以向管理员发送电子邮件，而不是重新运行该作业。在不脱离本发明的范围的情况下，可以开发和实现任何合适的自动化。也可以在特定时间运行UI自动化，或响应于用户请求或动作而运行UI自动化。

[0054] 在一些实施例中，提供专用的UI活动集合以自动化web功能。例如，识别web组件并与之交互的活动可以被定制为更准确地检测倾向于存在web应用中的形状和布局，这潜在地考虑了浏览器之间的视觉差异。

[0055] 在一些实施例中，可以在基于web的RPA设计器应用与基于桌面的RPA设计器应用之间提供集成。例如，可以以两个设计器应用版本都能读取的共用格式(诸如XAML)保存给定UI自动化的项目数据。这可以使RPA开发者能够在旅行的同时在他或她的笔记本电脑上离线处理UI自动化，在互联网接入可用时将项目上传到云，然后使用web浏览器继续处理UI自动化。在一些实施例中，也可以在设计器应用的web版本与桌面版本之间同步数据。此外，基于web的设计器应用的使用可以减少每个用户的本地存储器占用和处理需求，以及促进多个web用户之间的设计协作。

[0056] 在一些实施例中，自动化被设计和打包成由在VM、容器或操作系统会话中执行的RPA机器人执行，而无需修改特定平台的自动化包。换句话说，例如，相同的包可以在VMware®VM、Linux®容器或Windows®会话中运行，而无需修改包。可以针对特定目标平台设计RPA机器人并部署在其上，但是可以使用共用格式来运行自动化。例如，工作流程可以用XAML或一些其他格式来描述，每个平台上的RPA机器人可被设计成用编译的机器代码(例如，数字链接库(dll))来读取和执行这些其他格式。这可以提供代码混淆和安全性，以及潜在地使自动化运行得更快。然而，在某些实施例中，自动化可以是XAML、XML、纯文本或一些其他合适格式的脚本。

[0057] 图1是示出根据本发明实施例的基于web的RPA设计器系统100的架构图。RPA系统100包括web浏览器110(例如,运行在RPA开发者的计算系统上),web浏览器110允许开发者经由基于云的web应用120设计并实现RPA工作流程。web浏览器110经由web应用120可以提供针对应用集成以及自动化第三方应用、管理信息技术(IT)任务和业务IT流程的解决方案。Web浏览器110和web应用120可以促进自动化项目的开发,自动化项目包括流程的图表示。简而言之,web浏览器110和web应用120促进RPA工作流程的开发和自动化的部署,该自动化可以由作为服务运行的RPA机器人来实现。

[0058] 自动化项目通过让RPA开发者控制执行次序和工作流程中开发的自定义的一组步骤之间的关系(在本文定义为"活动"),实现了基于规则的流程的自动化。Web应用120的实施例的商业示例是UiPath Studio Web™。每个活动可以包括动作,诸如点击按钮、读取文件、写入日志面板等。在一些实施例中,工作流程可以被嵌套或嵌入。

[0059] 一些类型的工作流程可以包括但不限于序列、流程图、有限状态机(FSM)和/或全局异常处理程序。序列可以特别适合于线性流程,使流程能够从一个活动流向另一活动,而不会打乱工作流程。流程图可以特别适合于更复杂的业务逻辑,通过多个分支逻辑运算符,以更多样的方式实现决策的集成和活动的连接。FSM可以特别适合于大型工作流程。在其执行中,FSM可以使用有限数目的状态,这些状态是由条件(即,转换)或活动触发的。全局异常处理程序可以特别适合于在遇到执行错误时确定工作流程行为并且用于调试流程。

[0060] 一旦在web浏览器110中开发了工作流程,就可以使用web应用120来测试自动化的执行。然而,在一些实施例中,web浏览器110包括web应用120的自动化生成和测试功能。在自动化正常运行并准备部署在生产服务器140上之后,以机器可读代码或脚本的形式将自动化部署到生产服务器140的容器、VM或服务器操作系统会话。在一些实施例中,机器人130由指挥器150编排。RPA机器人130然后可以根据需要执行部署的自动化。可被管理的机器人130的类型包括但不限于有人值守机器人132、无人值守机器人134、开发机器人(类似于无人值守机器人134,但是用于开发和测试目的)和非生产机器人(类似于有人值守机器人132,但是用于开发和测试目的)。有人值守机器人132由用户事件触发,并且在相同计算系统上与人类一起操作。例如,有人值守机器人132可以执行UI自动化、提供可填写的表格等。有人值守机器人132可以帮助人类用户完成各种任务,并且可以由用户事件触发。在某些实施例中,有人值守机器人132只能从机器人托盘(robot tray)或者根据命令提示符启动。在一些实施例中,有人值守机器人132应当在人类监督下运行。

[0061] 无人值守机器人134可以无人值守地在虚拟环境、容器或操作系统会话中运行,并且可以自动化许多流程。无人值守机器人134可以负责例如远程执行、监测、调度以及为工作队列提供支持。在一些实施例中,对于所有机器人类型的调试可以在web浏览器110和web应用120中运行。有人值守和无人值守机器人都可以自动化各种系统和应用,包括但不限于大型机、web应用、VM、容器、企业应用(例如,由SAP®、SalesForce®、Oracle®等生产的应用)和计算系统应用(例如,台式电脑和膝上型电脑应用、移动设备应用、可穿戴计算机应用等)。

[0062] 机器人130是经由web应用120运行构建在web浏览器110中的工作流程的执行代理。(多个)机器人130的一些实施例的一个商业示例是UiPath Robots™。在一些实施例中,

机器人130默认安装Microsoft Windows®服务控制管理器(SCM)管理的服务。结果是,这种机器人130可以在本地系统帐户下打开交互式Windows®会话,并且具有Windows®服务的权限。

[0063] 在一些实施例中,可以以用户模式安装机器人130。对于这种机器人130,这意味着它们具有与安装了给定机器人130的用户相同的权限。该特征也可以适用于高密度(HD)机器人,这确保每个机器按其最大潜力进行充分利用。在一些实施例中,任何类型的机器人130可以被配置在HD环境中。

[0064] 执行器可以在Windows®会话下运行给定作业(即,它们可以执行工作流程)。执行器可以知道每显示器每英寸点数(DPI)设置。代理可以是在系统托盘窗口中显示可用作业的Windows® Presentation Foundation(WPF)应用。代理可以是服务的客户端。代理可以请求启动或停止作业以及改变设置。命令行是服务的客户端。命令行是可以请求启动作业并等待其输出的控制台应用。

[0065] 如上所述地划分机器人130的组件帮助开发者、支持用户,并且使计算系统更容易地运行、识别和跟踪每个组件正在执行什么。可以通过这种方式为每个组件配置特殊行为,诸如为执行器和服务建立不同的防火墙规则。在一些实施例中,执行器可以总是知道每个显示器的DPI设置。结果,工作流程可以按任何DPI执行工作流程,而不管在其上创建工作流程的计算系统的配置如何。在一些实施例中,来自web浏览器110和web应用120的项目也可以独立于浏览器缩放级别。在一些实施例中,针对不知道DPI或故意标记为不知道DPI的应用,可以禁用DPI。

[0066] 在一些实施例中,指挥器150可以具有各种能力,包括但不限于预置(provision)、部署、配置、排队、监测、日志记录和/或提供互连性。在某些实施例中,指挥器150可以被远程部署到生产服务器140。预置可以包括创建和维护机器人130与指挥器150之间的连接(例如,经由指挥器web应用)。部署可以包括确保将包版本正确递送给所指派的机器人130以用于执行。配置可以包括维护和递送机器人环境和流程配置。排队可以包括提供队列和队列项的管理。监测可以包括跟踪机器人标识数据以及维护用户许可。日志记录可以包括将日志存储和索引到数据库(例如,SQL数据库)和/或另一存储机制(例如,ElasticSearch®,它提供了存储和快速查询大型数据集的能力)。指挥器150可以通过充当第三方解决方案和/或应用的集中式通信点来提供互连性。指挥器150可以管理基础设施,基础设施包括所部署的机器人和所部署的机器人在其上进行操作的计算系统。

[0067] 图2是示出根据本发明实施例的计算系统200的架构图,计算系统200被配置为实现基于web的RPA设计器系统的部分或全部。在一些实施例中,计算系统200可以是本文描述和/或描绘的计算系统中的一个或多个。计算系统200包括用于传送信息的总线205或其他通信机制,以及耦合到总线205用于处理信息的(多个)处理器210。(多个)处理器210可以是任何类型的通用或专用处理器,包括中央处理单元(CPU)、专用集成电路(ASIC)、现场可编程门阵列(FPGA)、图形处理单元(GPU)、其多个实例和/或其任意组合。(多个)处理器210还可以具有多个处理核,并且至少一些核可以被配置为执行特定功能。在一些实施例中可以使用多并行处理。在某些实施例中,(多个)处理器210中的至少一个处理器可以是包括模拟

生物神经元的处理元件的神经形态电路。在一些实施例中，神经形态电路可以不要求冯·诺依曼计算架构的典型组件。

[0068] 计算系统200还包括用于存储信息和要由(多个)处理器210执行的指令的存储器215。存储器215可以由随机存取存储器(RAM)、只读存储器(ROM)、闪存、高速缓存、静态存储装置(诸如磁盘或光盘)的任意组合组成，或者是任何其他类型的非暂态计算机可读介质或其组合。非暂态计算机可读介质可以是可由(多个)处理器210访问的任何可用介质，并且可以包括易失性介质、非易失性介质、或者易失性介质和非易失性介质两者。介质也可以是可移动的，不可移动的，或者两者兼有。

[0069] 此外，计算系统200包括通信设备220(诸如收发器)，以经由无线和/或有线连接提供对通信网络的访问。在一些实施例中，通信设备220可以被配置为使用频分多址(FDMA)、单载波FDMA(SC-FDMA)、时分多址(TDMA)、码分多址(CDMA)、正交频分复用(OFDM)、正交频分多址(OFDMA)、全球移动通信系统(GSM)、通用分组无线业务(GPRS)、通用移动电信系统(UMTS)、cdma2000、宽带CDMA(W-CDMA)、高速下行链路分组访问(HSDPA)、高速上行链路分组访问(HSUPA)、高速分组访问(HSPA)、长期演进(LTE)、高级LTE(LTE-A)、802.11x、Wi-Fi、Zigbee、超宽带(UWB)、802.16x、802.15、家庭节点B(HnB)、蓝牙、射频识别(RFID)、红外数据协会(IrDA)、近场通信(NFC)、第五代(5G)、新无线电(NR)、其任意组合、和/或任何其他当前存在的或未来实现的通信标准和/或协议，而不脱离本发明的范围。在一些实施例中，通信设备220可以包括一个或多个天线，这些天线是单个的、阵列的、相控的、开关的、波束成形的、波束控制的、它们的组合、和/或任何其他天线配置，而不脱离本发明的范围。

[0070] (多个)处理器210还经由总线205耦合到显示器225，诸如等离子显示器、液晶显示器(LCD)、发光二极管(LED)显示器、场发射显示器(FED)、有机发光二极管(OLED)显示器、柔性OLED显示器、柔性基板显示器、投影显示器、4K显示器、高清晰度显示器、Retina®显示器、共面转换(IPS)显示器或用于向用户显示信息的任何其他合适的显示器。显示器225可以被配置为使用电阻、电容、表面声波(SAW)电容、红外、光学成像、色散信号技术、声脉冲识别、受抑全内反射等的触摸(触觉)显示器、三维(3D)触摸显示器、多输入触摸显示器、多触摸显示器等。在不脱离本发明的范围的情况下，可以使用任何合适的显示设备和触觉I/O。

[0071] 键盘230和光标控制设备235(诸如计算机鼠标、触摸板等)进一步耦合到总线205，以使用户能够与计算系统200交互。然而，在某些实施例中，可以不存在物理的键盘和鼠标，并且用户可以仅通过显示器225和/或触摸板(未示出)与设备交互。任何类型和组合的输入设备都可以用作设计选择。在某些实施例中，不存在物理的输入设备和/或显示器。例如，用户可以经由与其通信的另一计算系统与计算系统200远程交互，或者计算系统200可以自主操作。

[0072] 存储器215存储当由(多个)处理器210执行时提供功能的软件模块。这些模块包括用于计算系统200的操作系统240。这些模块还包括基于web的RPA模块245，RPA模块245被配置为执行本文描述的流程的全部或部分或其派生物。计算系统200可以包括一个或多个包括附加功能的附加功能模块250。

[0073] 本领域技术人员将理解，在不脱离本发明的范围的情况下，“系统”可以体现为服务器、嵌入式计算系统、个人计算机、控制台、个人数字助理(PDA)、蜂窝电话、平板计算设备、量子计算系统、或者任何其他合适的计算设备或设备组合。将上述功能呈现为由“系统”

执行并不旨在以任何方式限制本发明的范围，而是旨在提供本发明的许多实施例的一个示例。事实上，本文公开的方法、系统和装置可以以符合计算技术的本地化和分布式的形式实现，包括云计算系统。计算系统可以是局域网(LAN)、移动通信网络、卫星通信网络、互联网、公有云或私有云、混合云、服务器群、它们的任意组合等的一部分，或者可由它们访问。在不脱离本发明范围的情况下，可以使用任何本地化或分布式的架构。

[0074] 应当注意，本说明书中描述的一些系统特征已经被呈现为模块，以便更具体地强调它们的实现独立性。例如，模块可以被实现为硬件电路，包括定制的超大规模集成(VLSI)电路或门阵列、现成的半导体(诸如逻辑芯片、晶体管或其他分立元件)。模块也可以在可编程硬件设备中实现，诸如现场可编程门阵列、可编程阵列逻辑、可编程逻辑设备、图形处理单元等。

[0075] 模块也可以至少部分地在软件中实现，以供各种类型的处理器执行。所标识的可执行代码的单元可以例如包括计算机指令的一个或多个物理或逻辑块，这些物理或逻辑块可以例如被组织为对象、流程或函数。然而，所标识的模块的可执行文件不需要在物理上位于一起，而是可以包括被存储在不同位置的不同指令，当这些指令在逻辑上结合在一起时，构成该模块并实现该模块的所述目的。此外，在不脱离本发明的范围的情况下，模块可以被存储在计算机可读介质上，计算机可读介质可以是例如硬盘驱动器、闪存设备、RAM、磁带和/或用于存储数据的任何其他此类非暂时性计算机可读介质。

[0076] 实际上，可执行代码模块可以是单条指令或多条令，并且甚至可以分布在几个不同的代码段上、在不同的程序之间以及在跨几个存储设备。类似地，在文本中操作数据可以在模块内被标识出并被图示，并且可以以任何合适的形式被体现并且被组织在任何合适类型的数据结构中。操作数据可以作为单个数据集被收集，或者可以分布在包括不同的存储设备的不同的位置，并且可以至少部分地仅作为系统或网络上的电子信号而存在。

[0077] 图3A是示出根据本发明实施例的基于web的RPA系统300的架构图。系统300包括用户计算系统，诸如台式计算机302、平板电脑304和智能电话306。然而，在不脱离本发明范围的情况下，可以使用任何期望的计算系统，包括但不限于智能手表、膝上型计算机、物联网(IoT)设备、交通工具计算系统等。

[0078] 每个计算系统302、304、306具有安装在其上的web浏览器310。web浏览器310可以包括执行UI自动化(诸如执行鼠标点击、将文本输入到文本字段、提供截屏、提供出现在文本字段或网页的其他组件中的信息等)的本地部分的扩展。例如，这种扩展对于有人值守自动化可以是有用的。在某些实施例中，计算系统302、304、306可以运行已经使用基于web的RPA设计器系统开发的RPA机器人，该设计器系统从UI对象储存库中取回用于PowerPoint®、Outlook®等的元素。例如，参见美国专利申请第16/922,289号。用于目标应用的元素(例如，对象储存库内的项，诸如应用、屏幕和UI元素，其中应用对屏幕进行分组，并且屏幕对UI元素进行分组)可以被添加到工作流程(例如，通过将元素拖放到活动中)。在不脱离本发明的范围的情况下，web浏览器310可以是任何期望类型的web浏览器。

[0079] UI对象储存库可以包括UI对象的UI对象库，这些UI对象可以按照应用、应用版本、应用屏幕、UI元素的集合、它们的组合等进行分组。在一些实施例中，UI对象库的UI对象储存库可以促进管理、重用和增加项目中UI描述符的可靠性。为了使UI对象可重用，它们可以被提取到可供RPA流程引用的UI对象库中。例如，当选择器或其他UI描述符由于应用的新版

本而被修改时，库可以被重新创建(或重新发布)以包括修改后的UI描述符。然后，使用UI对象库的RPA流程可以调用修改后的UI描述符版本。

[0080] 来自UI对象库的UI描述符可以被直接添加到RPA工作流程活动中，这节省了开发者可能以其他方式为活动创建自定义选择器所需的时间。对象浏览器可以提供在对象库中存储创建的选择器的数据库，以实现UI描述符的可重用性。对象库在本文被定义为与来自某个版本的应用的一个或多个屏幕相对应的UI描述符的集合。UI描述符是用于寻找UI元素的指令集。

[0081] Web浏览器310运行web应用，并且经由网络320(例如，局域网(LAN)、移动通信网络、卫星通信网络、互联网、其任意组合等)向云环境中的服务器330提供与这些web应用相关的信息。服务器330运行包括自动化334和RPA机器人336的托管环境332(例如，容器、VM或操作系统会话)，自动化334和RPA机器人336经由扩展与web浏览器310交互和/或实现它们自己的与web浏览器310分离的(多个)流程。参见图6B。例如，自动化334和RPA机器人336可以在VM或容器内运行，并且在一些实施例中可以是无头的。在一些实施例中，服务器330可以将用于自动化334的数据(例如，不同版本的自动化334、对象储存库等)存储在数据库340中，并且加载它们或者以其他方式使它们可用于由RPA机器人336执行。

[0082] RPA机器人336与自动化334的不同之处在于，RPA机器人336是安装并运行在给定计算系统(诸如服务器330)上的机器侧服务。RPA机器人336可以等待来自指挥器应用(例如，UiPath Orchestrator™)、RPA设计器应用等的请求，然后在接收到请求之后运行所请求的自动化。相应的RPA机器人336接收机器可读代码或脚本的形式的自动化334的定义并将其运行。另一方面，自动化334是例如包形式的工作流程的实施例。在一些实施例中，包由RPA机器人336使用和运行。

[0083] 自动化334可以由RPA开发者使用运行在计算系统350上的基于web的RPA设计器应用352来设计。然而，在一些实施例中，基于web的RPA设计器应用远离计算系统350，并且经由web浏览器访问。RPA开发者可以创建或选择项目、创建或修改RPA工作流程、添加/修改/移除RPA工作流程活动等。在一些实施例中，基于web的RPA设计器应用352允许RPA开发者从UI对象储存库中选择元素，并且经由相应的UI描述符在RPA工作流程的活动中使用这些元素。一旦自动化334准备好由终端用户使用，由服务器330用作池的一部分，被部署在容器、VM或操作系统会话332中等，RPA开发者还可以使得自动化334在服务器330上被生成、执行测试并部署自动化334。在一些实施例中，自动化在计算系统350的web浏览器中本地生成，并且在那里被测试和验证，而不是远程地运行。

[0084] 图4A是示出根据本发明实施例的基于web的RPA开发应用网页400的主页视图的截屏。如果已经创建了各种样本项目，则可以将其显示给用户。RPA开发者可以通过点击新自动化按钮410来创建新自动化。在一些实施例中，可以呈现其他选项，诸如项目类型的集合，其可以包括但不限于流程、库项目、测试自动化项目、基于触发的流程等。用户还可以重新打开、编辑和重新发布已经创建的项目。

[0085] 当RPA开发者点击新自动化按钮410时，出现新自动化窗口420。参见图4B。用户可以输入自动化的名称和描述，并且通过点击创建按钮来创建新项目。然而，在不脱离本发明的范围的情况下，可以提供任何其他期望的信息，诸如(多个)开发者、目标公司等。

[0086] 在RPA开发者创建新自动化项目之后，出现画布430，自动化可以在画布430上被设

计。参见图4C。开始选项指定432允许用户指示自动化将如何启动。这里，自动化被配置为手动地启动。然而，用户可以将启动条件改变为另一选项，诸如基于触发(例如，用户点击网页上的按钮、某个网页被打开、web表单被提交等)、基于期望的启动时间、基于期望的周期运行时间(例如，每月的第一天午夜)等。用户可以通过点击空活动434来添加活动。

[0087] 当用户点击空活动434时，出现活动类别和快捷动作窗口440。参见图4D。用户可以从列出的类别中进行搜索或选择。例如，如果用户点击Excel®，则出现具有用于Excel®活动的选项的Excel®活动窗口450。参见图4E。

[0088] 如果用户选择使用Excel®文件，则创建使用Excel®文件活动460，一旦用户在点击“立即连接”按钮461之后提供连接信息，就可以配置使用Excel®文件活动460。对于某些应用，需要向基于web的RPA设计器应用、机器人托盘流程(例如，UiPath Assistant™)或其他服务授予许可，以访问应用和其他流程。凭据(credential)可以被保存并重用于将来的自动化。当RPA机器人执行自动化时，RPA机器人可以查看自动化所需的连接以及自动化将在哪些凭据下运行。RPA机器人可以被配置为在不改变底层工作流程逻辑的情况下访问这些凭据，并且如果找不到凭据或者凭据不能正常工作，则可以要求用户输入凭据。文件可以被托管在云中、被存储在远程数据库中等。活动的结束由结束使用Excel®文件指示符462指定。参见图4F。

[0089] 通过点击空活动464，可以在使用Excel®文件活动460中添加和配置其他活动、条件逻辑等。这可以调出与图4D的窗口440相同或相似的窗口。用以运行测试自动化的选项436和用以分享UI自动化项目的选项438也出现。这可以允许用户按照他们所创建的工作流程进行验证，并与其他人分享他们的工作。

[0090] 当用户配置活动时，出现各种选项。例如，如果用户点击“立即连接”按钮461并提供登录信息来访问远程文件和/或应用，则选项出现在选项窗格470的右侧。参见图4G。在这种情况下，用户从文件选择菜单471中选择Excel®文件“Apollo组件列表.xlsx”。文件名出现在文件名列表472中。用户可以使用测试活动菜单473以及配置通用选项474和基本选项475，测试活动的配置，这些选项是特定于给定活动的属性分组。还可以配置更具体的选项。例如，用户可以选择“在每个……上迭代(iterate over each)”复选框476，这允许用户选择电子表格中的元素，活动将在该元素上迭代。这里，用户已经用选项477选择了“行”和利用选项478选择了电子表格的“表1”。

[0091] 在一些实施例中，根据治理规则，用户可以被允许或不被允许发布或运行具有分析错误的自动化。在一些实施例中，验证错误是运行和发布自动化的先决条件，并且可以被许可用于所有用户。然而，静态代码分析错误可以是不被许可的，并且可以被这样标识。

[0092] 为了测试已经利用基于web的RPA设计器开发的自动化(诸如图4A-图4G中所示)，执行可以发生在服务器侧(例如，在云中)，而不是在开发者的计算系统上。除了设计自动化并将其执行之外，RPA开发者可以不需要进一步的配置。执行可以发生在远程VM上，远程VM可以由也可以不由指挥器应用(诸如UiPathOrchestrator®)中的RPA开发者来配置。可以经由基于web的RPA设计器的web浏览器部分向RPA开发者通知成功执行，或者向其提供执行

错误和/或违反治理规则的情况。在某些实施例中，在自动化被批准和发布供生产使用(即，运行时使用)之前，RPA工作流程可以由具有适当许可的个人进行审查。

[0093] 应当注意，图4A-图4G仅作为示例提供，并且在不脱离本发明的范围的情况下，可以配置各种其他活动和条件。例如，如果存在不同的条件，则RPA开发者可以选择对Excel®文件执行不同动作(例如，如果组件的截止日期已过，则发送关于该截止日期的电子邮件，相对于如果该组件未过期，则发送具有新的最终组件的电子邮件并写入Excel®中更新状态的单元格，在其他非电子邮件应用中发送电子邮件和/或消息等)。在不脱离本发明的范围的情况下，可以使用一些实施例的基于web的RPA设计器来设计自动化，该设计器可以与网页交互、从网页获得信息、以及在服务器侧执行期望的RPA机器人动作，这些动作可以包括与用户网页交互、与服务器侧或用户计算系统本地的其他RPA机器人交互、从数据库取回信息、或者任何其他RPA流程。如上所述，在某些实施例中，自动化可以经由VM或容器在服务器侧被部分或全部执行。

[0094] 图5是示出根据本发明实施例的用于在设计时执行基于web的RPA开发的流程的流程图500。该流程开始于510，在web浏览器上访问并显示基于web的设计器应用的web界面。Web界面允许用户创建RPA项目、配置RPA工作流程、以及向服务器侧的基于web的RPA设计器应用提交RPA工作流程配置，以远程生成、执行和验证自动化。在520处，基于web的设计器应用经由web界面设计并配置RPA工作流程。

[0095] 在用户对RPA工作流程满意之后，或者当用户另外想要执行测试和验证时，在530处，用户经由web界面使得服务器侧的基于web的RPA设计器应用生成实现工作流程的自动化。然后，在540处，服务器侧的基于web的RPA设计器应用经由RPA机器人执行自动化，并验证自动化。如果在550处验证失败，则用户可以在520继续设计并配置RPA工作流程。如果验证成功，则可以在560处发布RPA工作流程/项目用于生产(运行时)使用。例如，这可以涉及生成自动化代码以及经由VM、容器或操作系统会话中的RPA机器人运行自动化代码。在一些实施例中，该流程然后可以进行到图6、图7或图9的步骤。

[0096] 图6是示出根据本发明实施例的用于在运行时执行与web扩展交互的自动化的流程600的流程图。该流程开始于610处，经由RPA机器人执行服务器侧自动化。服务器侧自动化可以被调度、手动执行、基于触发执行等。触发可以是应用事件、用户与网页的交互发生、经过某个时间等。然后，在620处，RPA机器人向web浏览器的web扩展(例如，在客户端计算系统上、在VM中、在容器中等)发送一个或多个命令，以与web浏览器交互，从web浏览器的一个或多个标签中的一个或多个网页获得信息，或者两者兼有。在630处，如RPA机器人所请求的，web扩展然后与web浏览器交互，从web浏览器的一个或多个标签中的一个或多个网页获得信息，或者既与web浏览器交互，又从web浏览器的一个或多个标签中的一个或多个网页获得信息。在一些实施例中，web扩展提供本地功能，例如，当在容器中运行自动化并且作为用该扩展进行定义的结果时，可以无头地执行该本地功能。在一些实施例中，这种功能可以针对台式/膝上型计算机、瘦客户端、胖客户端等来实现。可以从本地计算系统获得信息和/或可以在本地执行浏览器交互。

[0097] 然后，在640处，web扩展提供对所请求的交互已发生的确认，提供所请求的信息，或者如果存在问题，则提供错误消息，RPA机器人接收web扩展所提供的这些。在一些实施例

中，这可以是自动化被设计来实现的任务。然而，如果自动化被设计成基于来自web扩展的信息执行一些附加处理或其他动作，诸如执行数据库查找、发送电子邮件、写入和保存文件、请求来自web扩展的附加动作和/或信息等，则RPA机器人可以在650处执行这些动作。对于一个或多个其他命令，该流程也可以返回到步骤620。

[0098] 图7是示出根据本发明实施例的用于在运行时执行运行在VM或容器中的自动化的流程700的流程图。该流程开始于710，创建实现相应自动化的VM和/或容器的池。例如，池可以是弹性预置的或手动预置的。然后，在720处，由RPA机器人在VM、容器或操作系统会话中执行服务器侧自动化。服务器侧自动化可以被调度、被手动执行、被基于触发器执行等。然后，在730处，RPA机器人执行在自动化工作流程中设计的动作。

[0099] 在一些实施例中，基于web的设计器系统可以被用于设计自动化，该自动化经由本地RPA扩展流程与运行在计算系统上的应用的本地实例交互。本地RPA扩展流程可以从远程运行在容器、VM或操作系统会话中的RPA机器人接收通信，与本地应用交互，并且将信息发送回发送请求的RPA机器人。图8是示出根据本发明实施例的被配置为与本地应用交互的基于web的远程流程控制系统800的架构图。类似于图3A和图3B的系统300，系统800包括台式计算机802、平板电脑804、智能电话806、网络820、服务器830、托管环境832、数据库840、开发者计算系统850和基于web的RPA设计器应用852。同样类似于系统300，托管环境832包括自动化834和RPA机器人836。

[0100] 然而，在系统800中，台式计算机802、平板电脑804和智能电话806运行本地RPA扩展流程810，本地RPA扩展流程810扩展RPA机器人836的功能以与运行在台式计算机802、平板电脑804和智能电话806上的应用交互。本地RPA扩展流程810可以经由网络820从执行自动化834的RPA机器人836接收请求。例如，本地RPA扩展流程810可能正在监听端口上的通信。在不脱离本发明范围的情况下，可以使用任何合适的通信协议(例如，传输控制协议(TCP)/互联网协议(IP)、文件传输协议(FTP)等)。

[0101] 本地RPA扩展流程810可以解析来自RPA机器人836的请求，并执行相关联的动作和/或取回所请求的信息。例如，来自RPA机器人836的请求可以要求本地RPA扩展流程810使用Outlook®的API来搜索用户的电子邮件中的术语“发票”，取回匹配结果，并将结果发送回RPA机器人836。RPA机器人836然后可以使用该信息来继续执行自动化834的逻辑。

[0102] 在一些实施例中，本地RPA扩展流程810可以使用流程间通信(IPC)协议与在相应计算系统上运行的其他流程通信。IPC协议是一种这样的机制，通过该机制，运行在操作系统中的流程可以与同样运行在该操作系统上(可能在不同的会话中)的其他流程通信。这些协议可以促进经由网络820、管道、组件对象模型(COM)、远程流程调用(RPC)、套接字等的通信。IPC协议可以用于在运行的流程之间发送信息、请求、命令等。

[0103] 图9是示出根据本发明实施例的用于在运行时执行与本地RPA扩展流程交互的自动化的流程900的流程图，该本地RPA扩展流程运行在不同于RPA机器人的计算系统上。该流程开始于910，经由RPA机器人执行服务器侧自动化。服务器侧自动化可以被调度、被手动执行、基于触发被执行等。触发可以是应用事件、用户与网页的交互发生、经过某个时间等。然后，在920处，RPA机器人向运行在远程计算系统上(例如，在客户端侧计算系统上、在远程VM中、在远程容器中等)的本地RPA扩展流程发送一个或多个请求，以与对于本地RPA扩展流程是本地运行的一个或多个应用和/或流程(例如，操作系统流程、与应用相关联的流程等)交

互。然后，在930处，本地RPA扩展流程控制一个或多个应用(例如，经由API)和/或流程以在(多个)应用中执行期望的操作，从(多个)应用和/或计算系统上的其他流程获得信息，或者两者兼有。然后，在940处，本地RPA扩展应用提供对所请求的与(多个)应用的(多个)交互已发生的确认，提供所请求的信息，或者提供所请求的与(多个)应用的(多个)交互已发生的确认和所请求的信息两者。在一些实施例中，如果存在问题，则可以提供RPA机器人接收的错误消息。

[0104] 在一些实施例中，这可以是自动化被设计来实现的任务。然而，如果自动化被设计成基于来自web扩展的信息执行一些附加处理或其他动作(诸如执行数据库查找、发送电子邮件、写入和保存文件、请求来自web扩展的附加动作和/或信息等)，则在950处，RPA机器人可以执行这些动作。对于一个或多个其他请求，该流程也可以返回到步骤920。

[0105] 图10是示出根据本发明实施例的用于在设计时经由web浏览器执行基于web的RPA开发而无需RPA机器人(即，无机器人(robotless))的流程1000的流程图。本文使用的“无机器人”意味着类似于RPA机器人的功能被包括在web浏览器中，但是RPA机器人不是作为单独的服务而运行的。因此，不需要在运行web浏览器的计算系统上安装本地客户端侧二进制文件，并且在一些实施例中只需要web浏览器。类似于机器人的二进制文件可以被包括在标签内，并在加载时从web服务器提供服务。HTML、JavaScript™等也可以在那时被加载。

[0106] 该流程开始于1010处，将用于基于web的RPA设计器应用的代码下载到web浏览器。代码允许用户既创建自动化又执行自动化。该代码可以包括加载到web浏览器的标签中的可执行二进制文件(诸如DLL)。这不是典型的，因为通常加载其他格式(诸如JavaScript™)。然而，应当注意，在不脱离本发明的范围的情况下，可以使用包括自动化执行功能(诸如JavaScript™、Servlets™等)的任何合适的代码。在1020处，基于web的RPA设计器应用的web界面被显示在web浏览器中。web界面被配置为提供用于创建RPA项目和配置RPA工作流程以及由此生成和执行自动化的功能。在1030处，基于web的设计器应用经由web界面设计和配置RPA工作流程。

[0107] 在用户对RPA工作流程满意之后，或者当用户另外想要执行测试和验证时，在1040处，web浏览器为配置的RPA工作流程生成自动化。这种自动化可以作为浏览器工作流程而不是规则浏览器流程来运行。然后，在1050处，web浏览器执行并验证生成的自动化。如果在1060处验证失败(例如，存在错误或违反治理规则)，则用户可以在1030处继续设计和配置RPA工作流程。如果验证成功，则可以在1070处发布RPA工作流程/项目以用于生产(运行时)使用。例如，这可以涉及为web浏览器生成自动化代码以及在VM、容器或操作系统会话中运行web浏览器的自动化代码。在一些实施例中，该流程然后可以进行到图11的步骤。

[0108] 在该实施例中，用于执行和调试自动化的机制被加载在web浏览器本身中。例如，这不要求远程VM、容器或操作系统会话。换句话说，当用户运行自动化时，代码不会被转移到远程容器、远程VM或远程操作系统会话中执行。相反，自动化设计时间和运行时功能处于web浏览器中，在一些实施例中可能在相同的标签中。当用于基于web的RPA设计器应用的网页被加载时，可以在web浏览器中加载用于执行自动化的功能。在一些实施例中，这不要求web浏览器扩展。因此，标签本身本质上变成了设计时RPA设计器和运行时RPA机器人两者，并且用户可以创建、执行和调试自动化。在一些实施例中，自动化也可以被设计成在生产服务器的web浏览器中运行，而不是由生产RPA机器人执行。

[0109] 图11是示出根据本发明实施例的用于在运行时在容器、VM或操作系统会话的web浏览器中执行自动化的流程1100的流程图。该流程开始于1110，经由运行在服务器的容器、VM或操作会话中的web浏览器执行服务器侧自动化。光是这种自动化形式就是新颖的。然而，在一些实施例中，在1120处，自动化向运行在远程计算系统上的web扩展或本地RPA扩展流程发送命令，这分别类似于图6和图8的流程600和800。web扩展或本地RPA扩展流程然后在1130处执行所请求的动作和/或获得所请求的信息，并且在1140处向服务器的容器、VM或操作会话中的web浏览器提供确认和/或信息。然后，在1150处，web浏览器可以执行与自动化相关联的附加动作。

[0110] 根据本发明的实施例，图5-图7和图9-图11中执行的流程步骤可以由计算机程序执行，该计算机程序编码用于使(多个)处理器执行图5-图7和图9-图11中描述的(多个)流程的至少一部分的指令。计算机程序可以体现在非暂态计算机可读介质上。计算机可读介质可以是但不限于硬盘驱动器、闪存设备、RAM、磁带、和/或用于存储数据的任何其他这种介质或介质组合。计算机程序可以包括用于控制计算系统的(多个)处理器(例如，图2的计算系统200的(多个)处理器210)来实现图5-图7和图9-图11中描述的全部或部分流程步骤的编码指令，这些指令也可以被存储在计算机可读介质上。

[0111] 计算机程序可以用硬件、软件或混合实现来实现。计算机程序可以由彼此有效通信并且被设计成传递信息或指令以进行显示的模块组成。计算机程序可以被配置为在通用计算机、ASIC或任何其他合适的设备上操作。

[0112] 容易理解，本发明的各种实施例的组件(如本文的附图中总体上描述和示出的)可以以各种各样不同的配置被布置和设计。因此，本发明实施例的详细描述(如附图中所示)不旨在限制所要求保护的本发明的范围，而仅仅是本发明的选定实施例的代表。

[0113] 贯穿本说明书描述的本发明的特征、结构或特性可以以任何合适的方式组合在一个或多个实施例中。例如，贯穿本说明书对“某些实施例”、“一些实施例”或类似语言的引用意味着结合该实施例描述的特定的特征、结构或特性被包括在本发明的至少一个实施例中。因此，贯穿本说明书的短语“在某些实施例中”、“在一些实施例中”、“在其他实施例中”或类似语言的出现不一定都指同一组实施例，并且所描述的特征、结构或特性可以以任何合适的方式组合在一个或多个实施例中。

[0114] 应当注意，贯穿本说明书对特征、优点或类似语言的引用并不意味着可用本发明实现的所有特征和优点应当是本发明的任何单个实施例，或者在本发明的任何单个实施例中。相反，涉及特征和优点的语言被理解为意味着结合实施例描述的特定的特征、优点或特性被包括在本发明的至少一个实施例中。因此，贯穿本说明书的对特征和优点的讨论以及类似语言可以但不一定是指相同的实施例。

[0115] 此外，所描述的本发明的特征、优点和特性可以以任何合适的方式结合在一个或多个实施例中。相关领域的技术人员将认识到，本发明可以在没有特定实施例的一个或多个特定特征或优点的情况下被实践。在其他情况下，在某些实施例中可以认识到可能不存在于本发明的所有实施例中的附加特征和优点。

[0116] 本领域普通技术人员将容易理解，可以用不同次序的步骤和/或用与所公开的配置不同的硬件元件来实践上述本发明。因此，尽管已经基于这些优选实施例描述了本发明，但是对于本领域技术人员来说显而易见的是，在保持在本发明的精神和范围内的同时，某

些修改、变化和替代构造将是显而易见的。因此，为了确定本发明的边界和界限，应当参考所附权利要求。

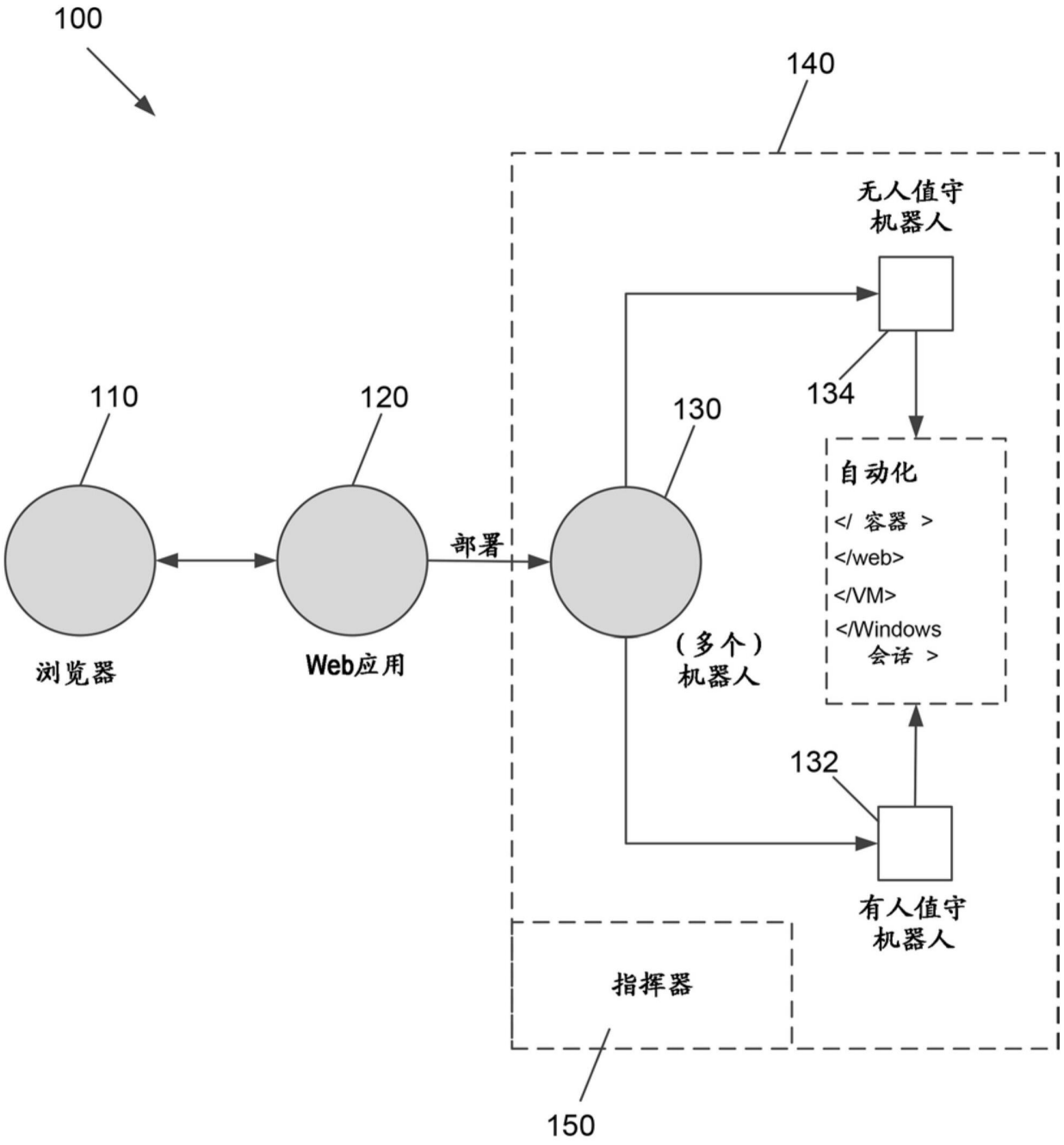

100
140
无人值守
机器人
134
110
120
130
自动化
</ 容器 >
</web>
</VM>
</Windows
会话 >
部署
浏览器
Web应用
(多个)
机器人
132
有人值守
机器人
指挥器
150

图1

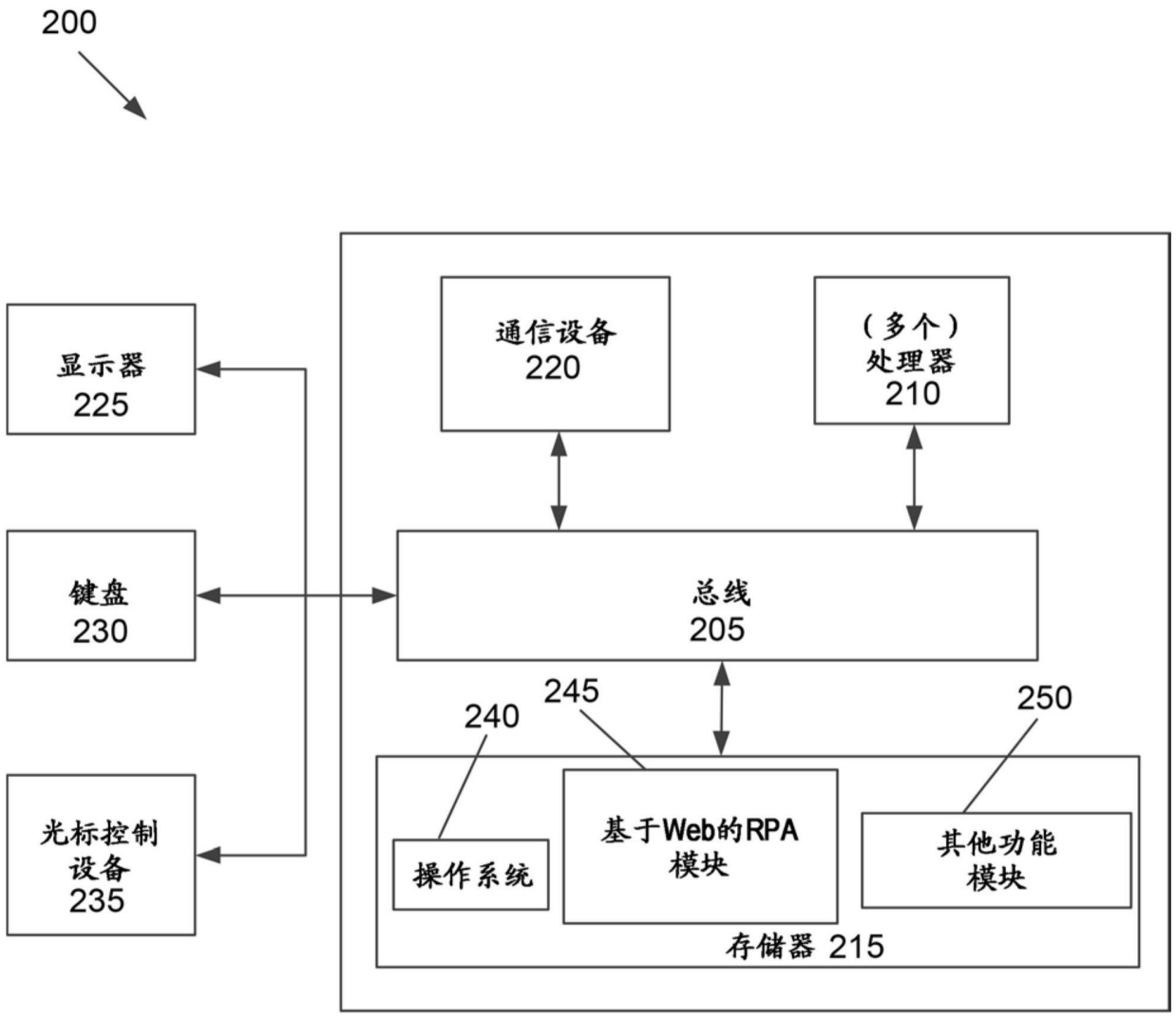

200
显示器
225
键盘
230
光标控制
设备
235
通信设备
220
(多个)
处理器
210
总线
205
240
245
250
操作系统
基于Web的RPA
模块
其他功能
模块
存储器 215

图2

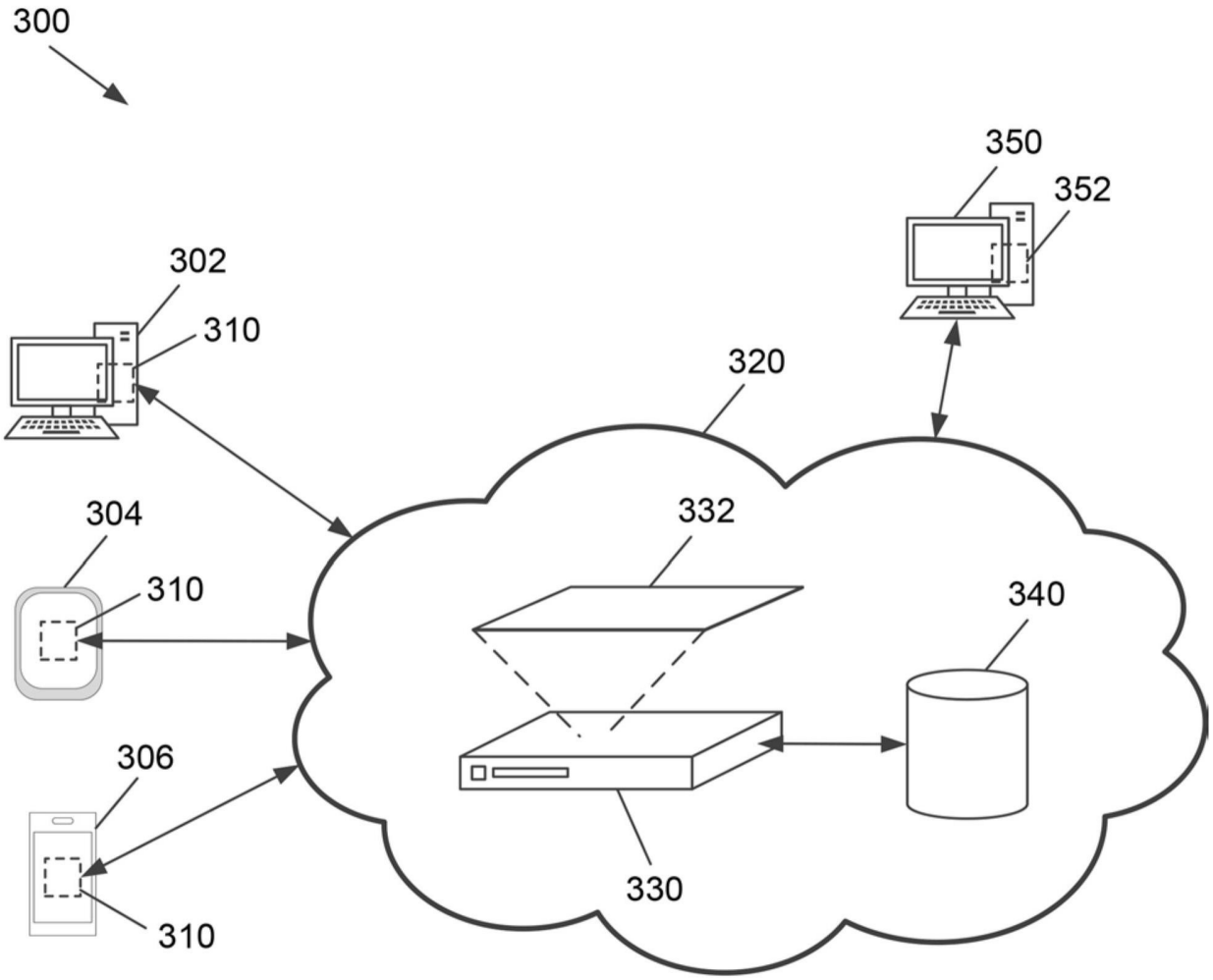


图3A

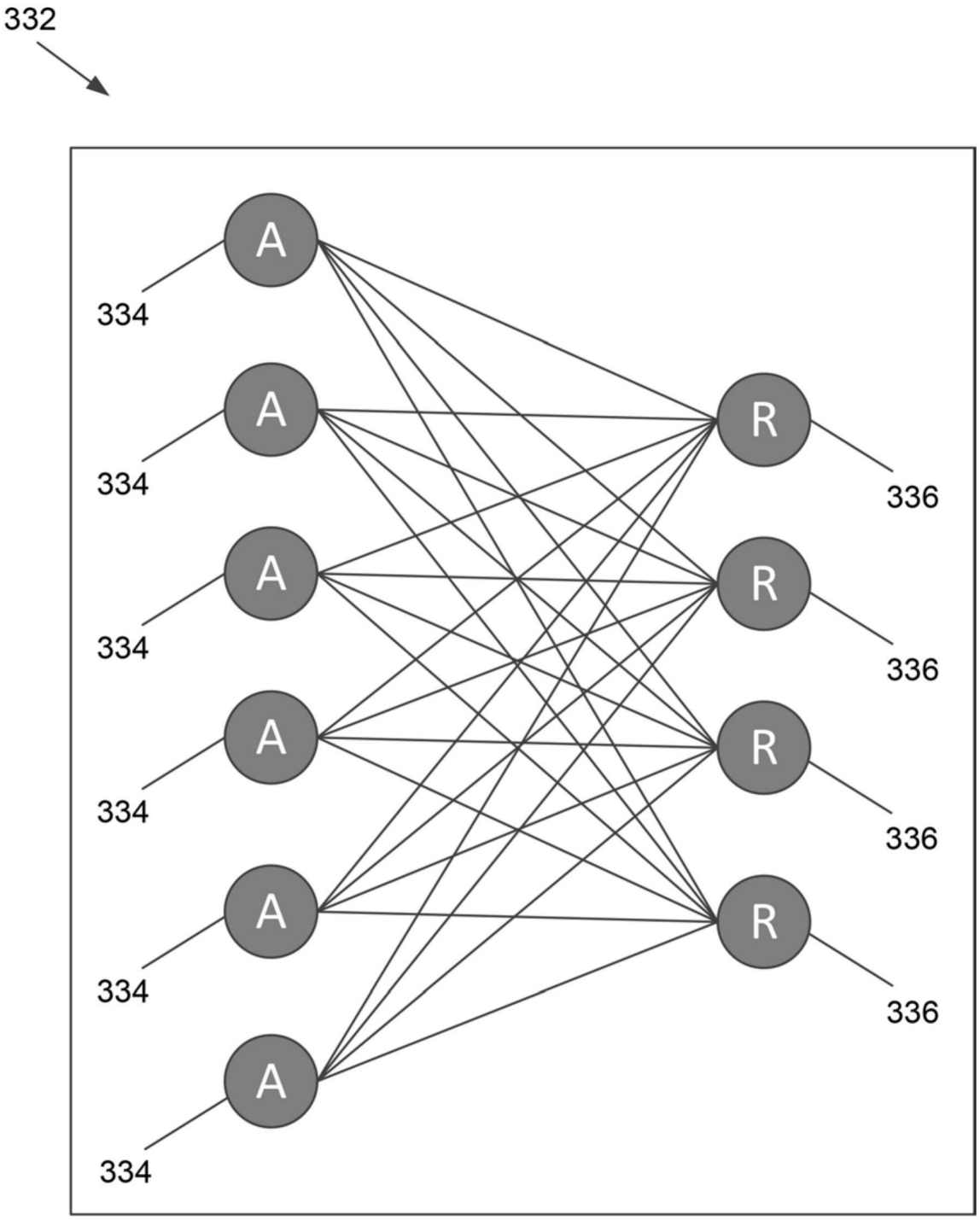


图3B

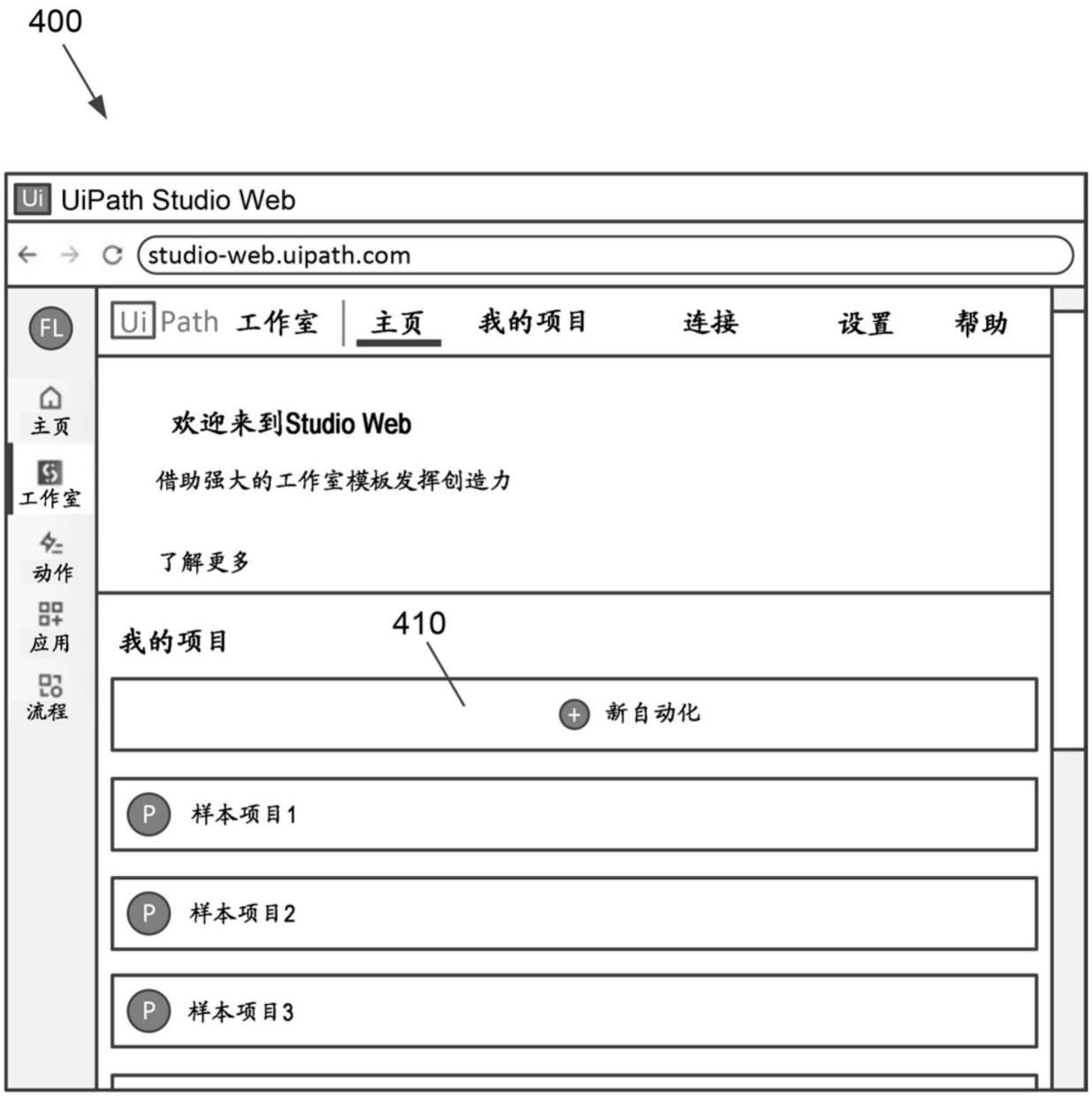

400
UiPath Studio Web
studio-web.uipath.com
FL
UiPath 工作室
主页
我的项目
连接
设置
帮助
主页
工作室
动作
应用
流程
欢迎来到Studio Web
借助强大的工作室模板发挥创造力
了解更多
我的项目
410
新自动化
样本项目1
样本项目2
样本项目3

图4A


400
UiPath Studio Web
studio-web.uipath.com
FL
UiPath 工作室
主页
我的项目
连接
设置
帮助
主页
工作室
动作
应用
流程
420
欢迎来到Studio Web
借助强
了解更
新自动化
给您的自动化命名并描述。如果您改变主意，你可以之后再改变这些。
名称 *
我的自动化
描述
写描述来帮助人们理解这个自动化解决了什么
创建
取消
我的项目
样本
样本
样本项目3

图4B

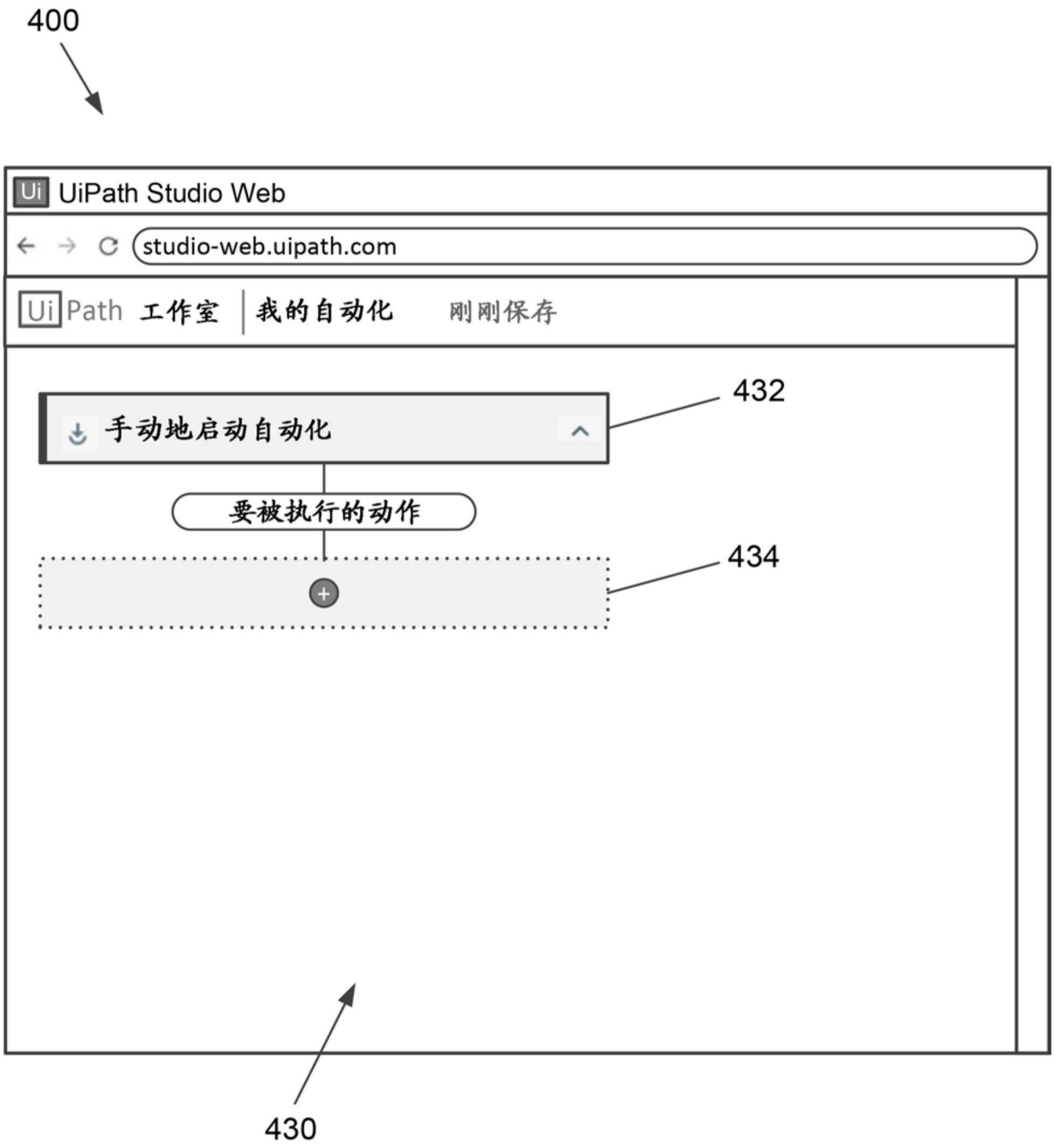

400
UiPath Studio Web
studio-web.uipath.com
UiPath 工作室
我的自动化
刚刚保存
手动地启动自动化
要被执行的动作
432
434
430

图4C

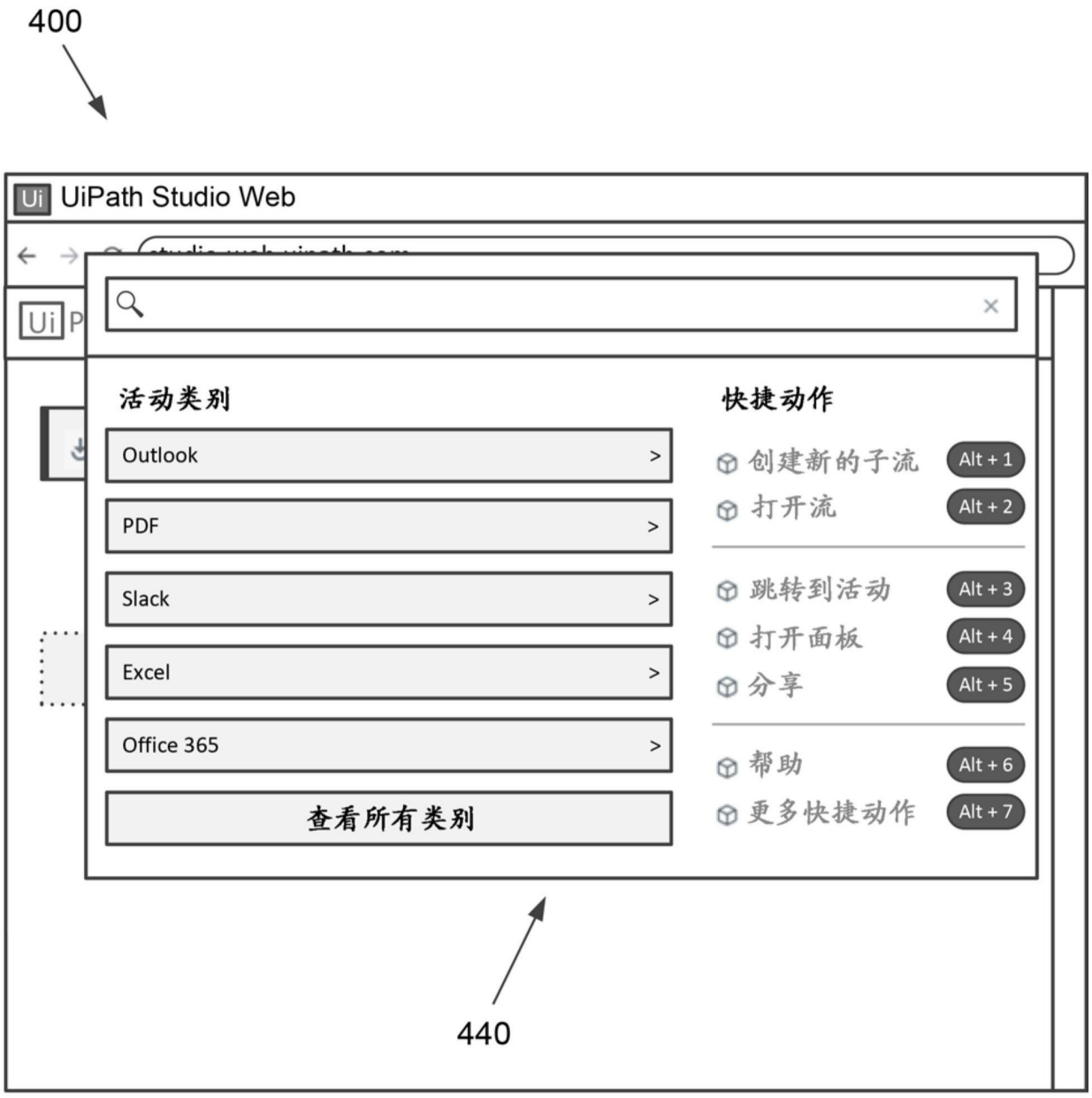

400
UiPath Studio Web
活动类别
Outlook
PDF
Slack
Excel
Office 365
查看所有类别
快捷动作
创建新的子流 Alt + 1
打开流 Alt + 2
跳转到活动 Alt + 3
打开面板 Alt + 4
分享 Alt + 5
帮助 Alt + 6
更多快捷动作 Alt + 7
440

图4D

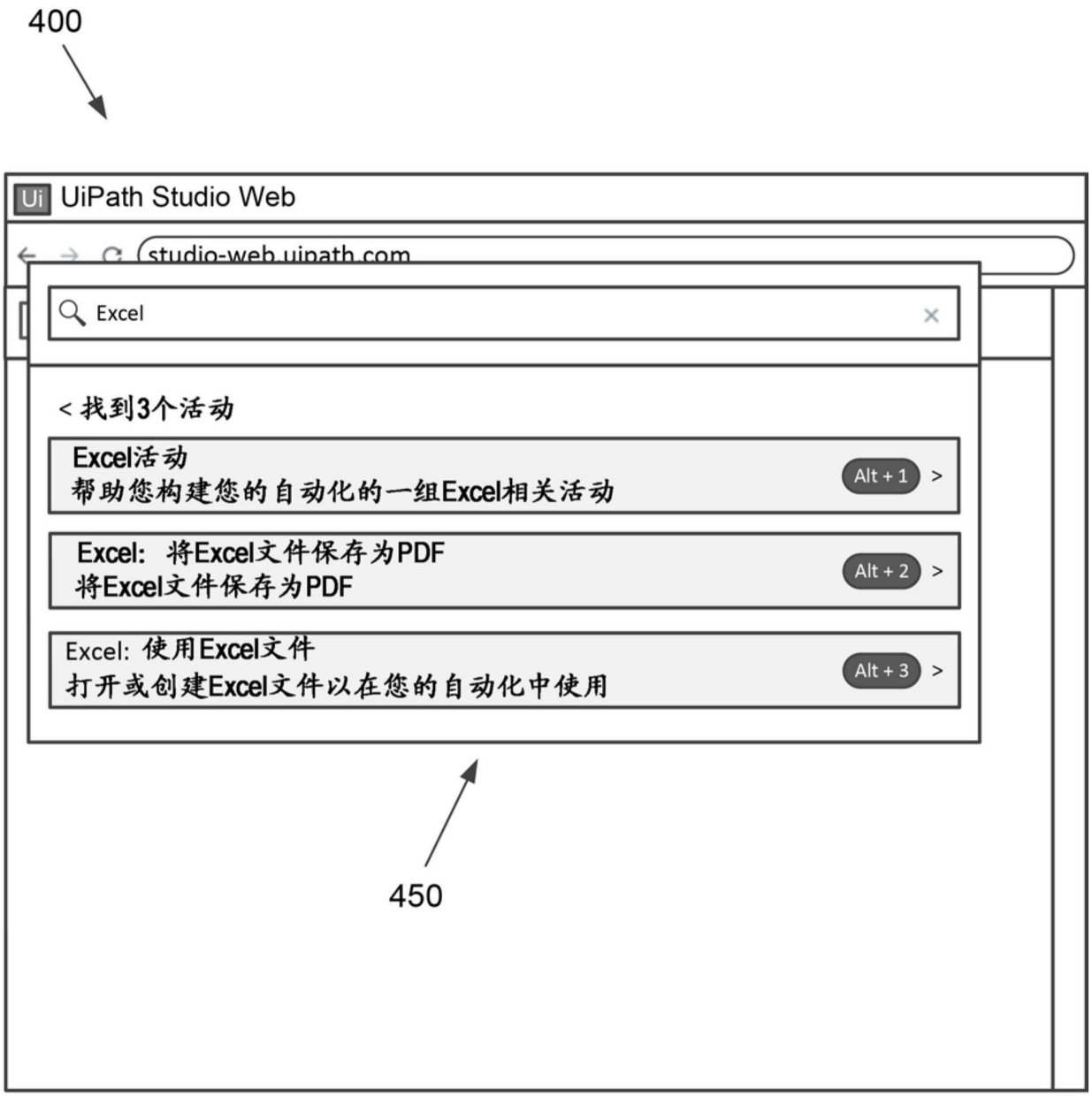

400
UiPath Studio Web
studio-web.uipath.com
Excel
< 找到3个活动
Excel活动
帮助您构建您的自动化的一组Excel相关活动
Alt + 1
Excel: 将Excel文件保存为PDF
将Excel文件保存为PDF
Alt + 2
Excel: 使用Excel文件
打开或创建Excel文件以在您的自动化中使用
Alt + 3
450

图4E

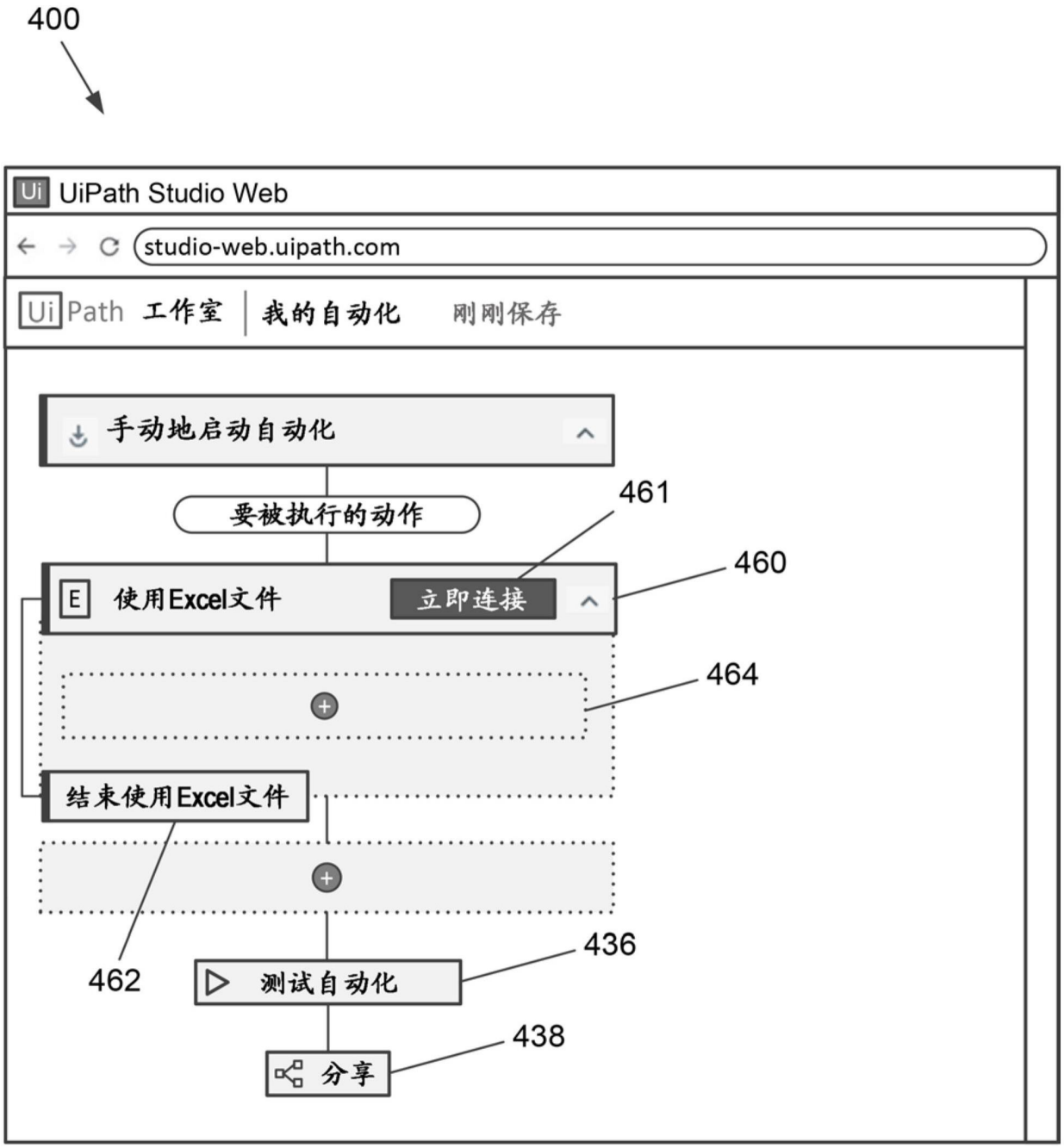

400
UiPath Studio Web
studio-web.uipath.com
UiPath 工作室
我的自动化
刚刚保存
手动地启动自动化
要被执行的动作
461
460
使用Excel文件
立即连接
464
结束使用Excel文件
462
436
测试自动化
438
分享

图4F

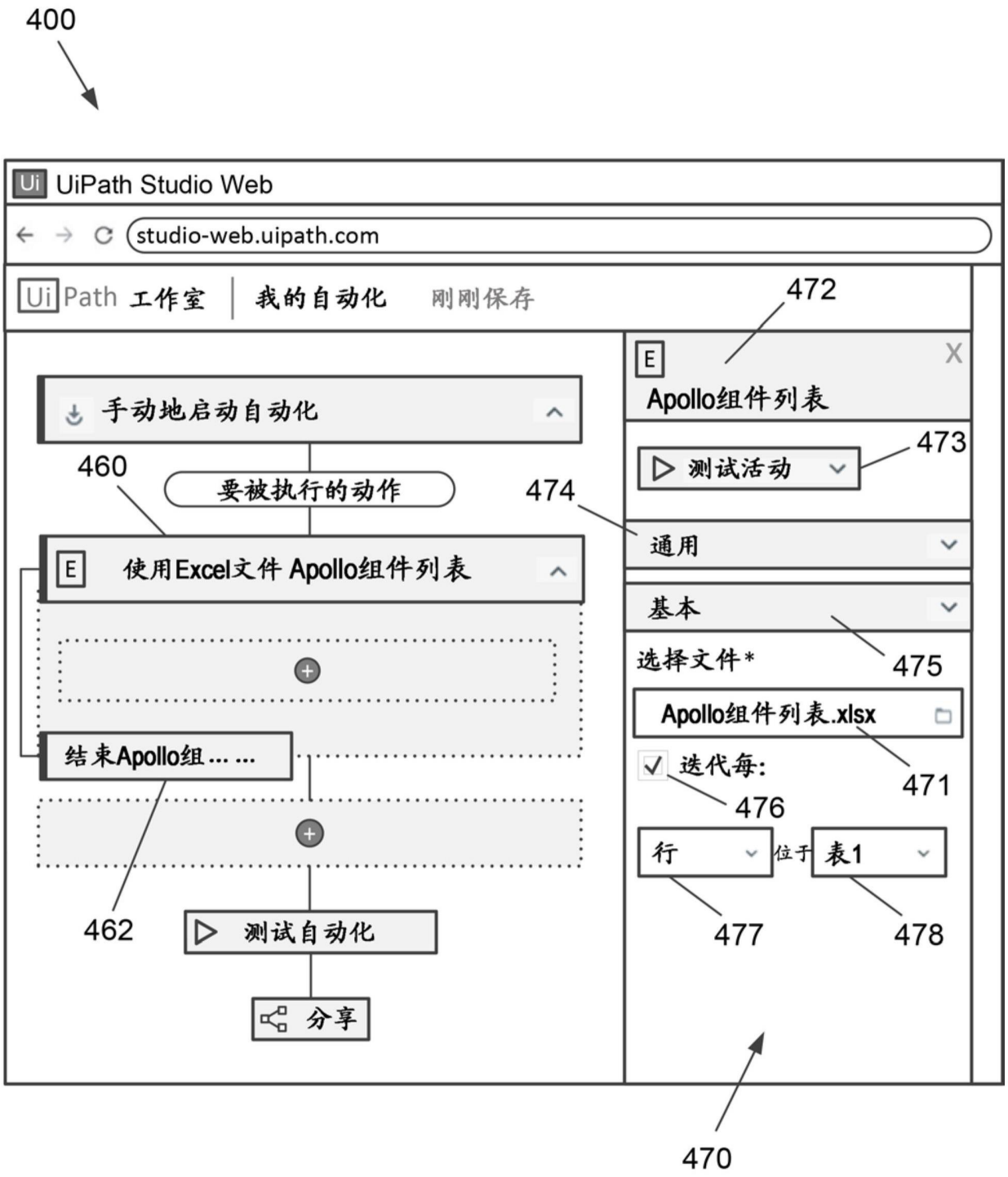

400
UiPath Studio Web
studio-web.uipath.com
UiPath 工作室
我的自动化
刚刚保存
472
手动地启动自动化
460
要被执行的动作
474
使用Excel文件 Apollo组件列表
结束Apollo组... ...
462
测试自动化
分享
Apollo组件列表
473
测试活动
通用
基本
选择文件*
475
Apollo组件列表.xlsx
471
迭代每:
476
行
位于
表1
477
478
470

图4G

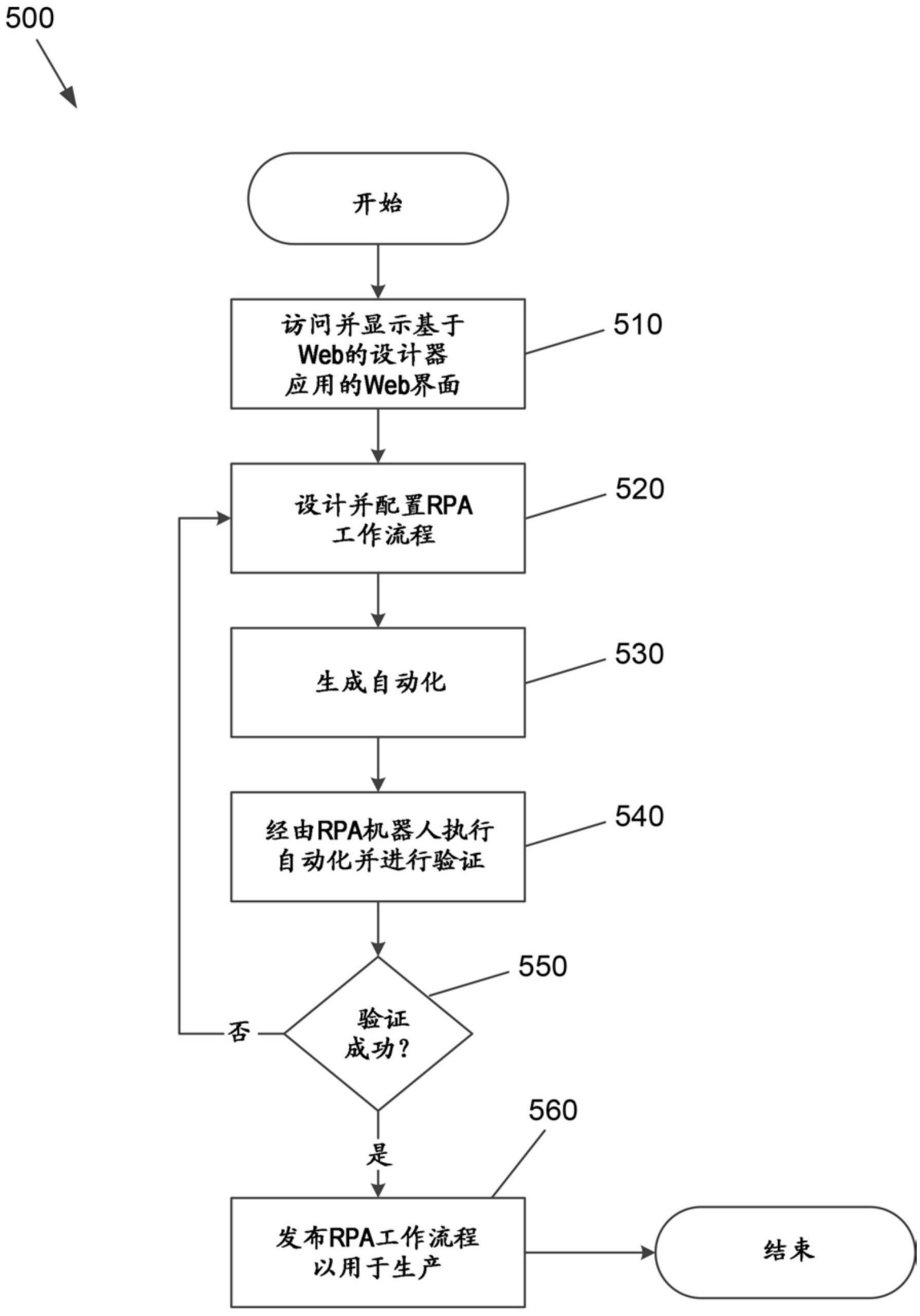

500
开始
访问并显示基于Web的设计器应用的Web界面
510
设计并配置RPA工作流程
520
生成自动化
530
经由RPA机器人执行自动化并进行验证
540
验证成功?
550
否
是
560
发布RPA工作流程以用于生产
结束

图5

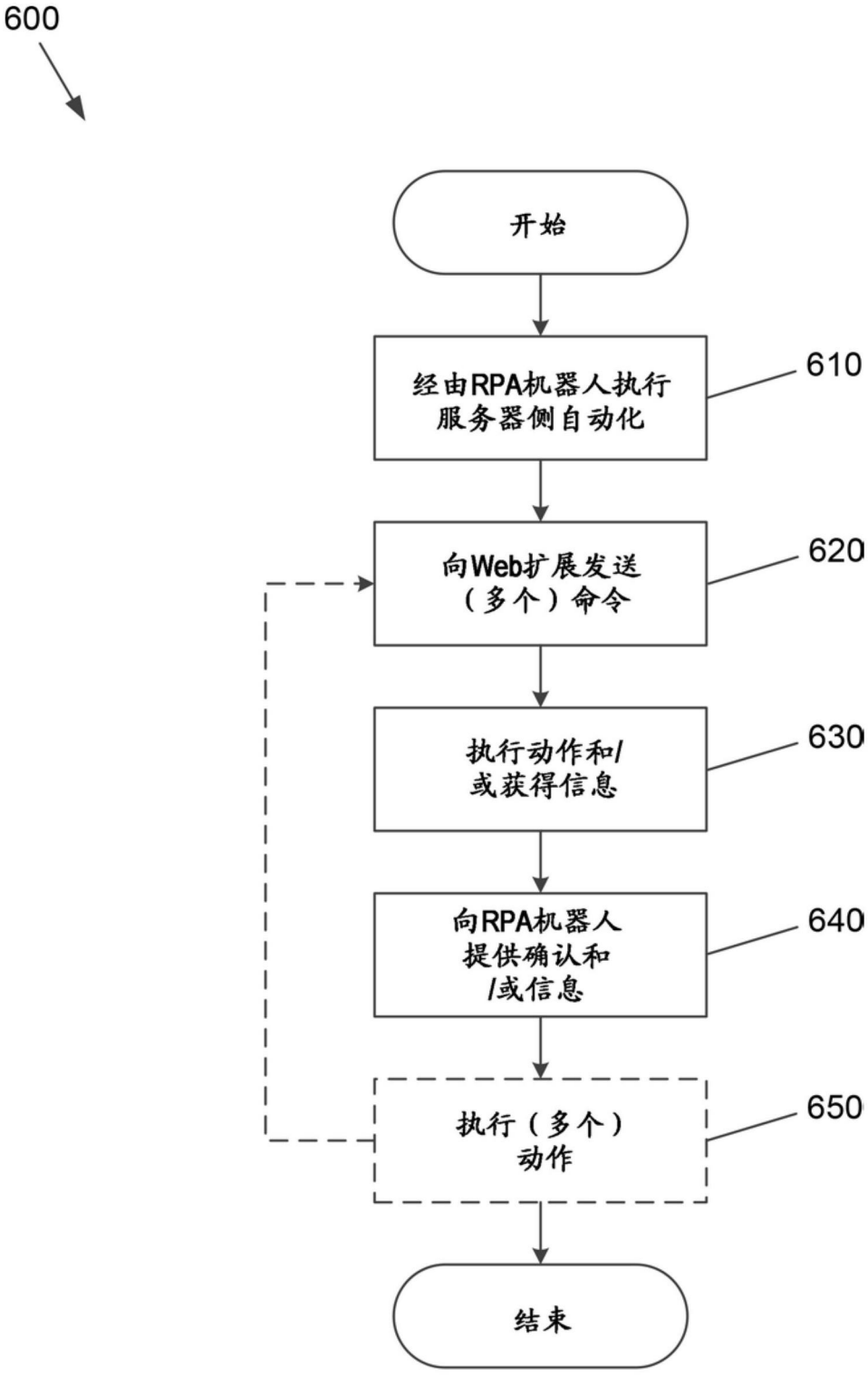

600
开始
经由RPA机器人执行
服务器侧自动化
610
向Web扩展发送
（多个）命令
620
执行动作和/
或获得信息
630
向RPA机器人
提供确认和
/或信息
640
执行（多个）
动作
650
结束

图6

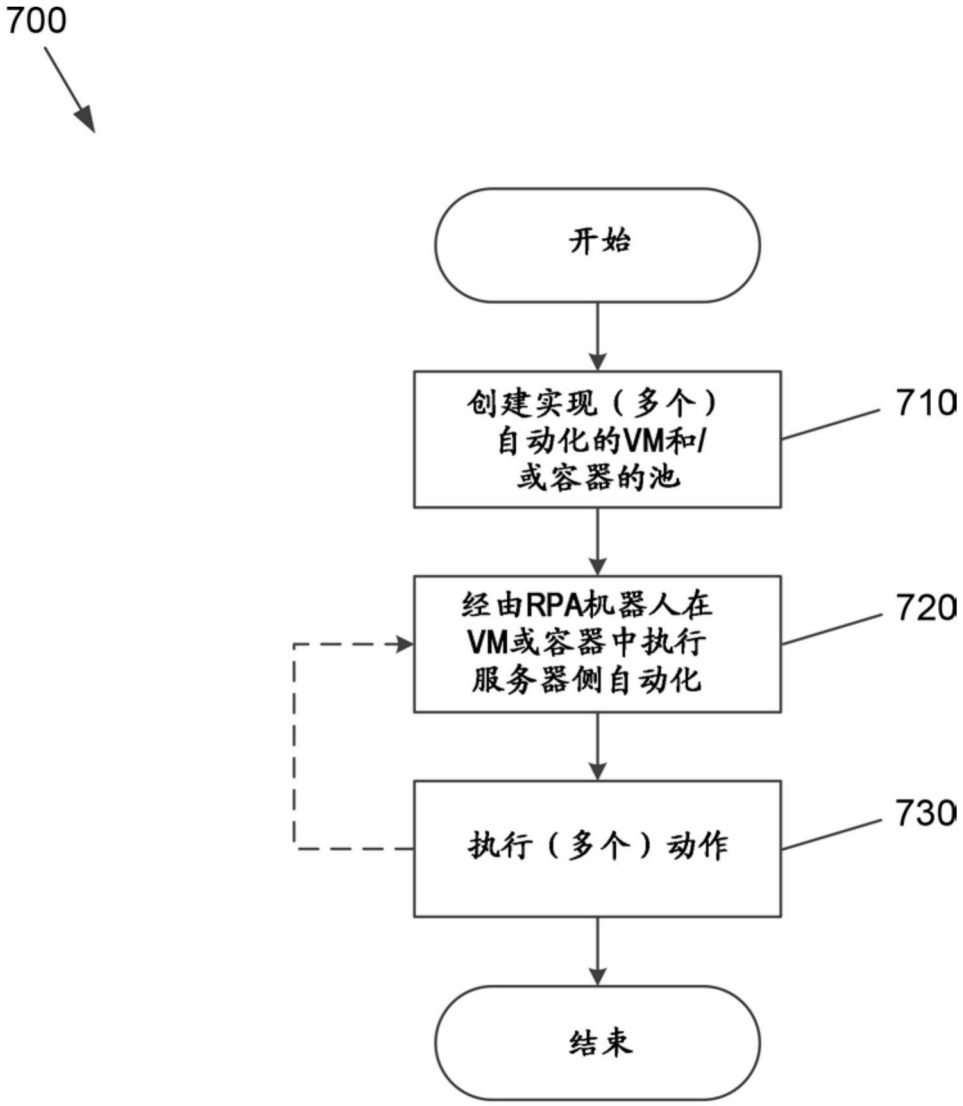

700
开始
创建实现（多个）自动化的VM和/或容器的池
710
经由RPA机器人在VM或容器中执行服务器侧自动化
720
执行（多个）动作
730
结束

图7

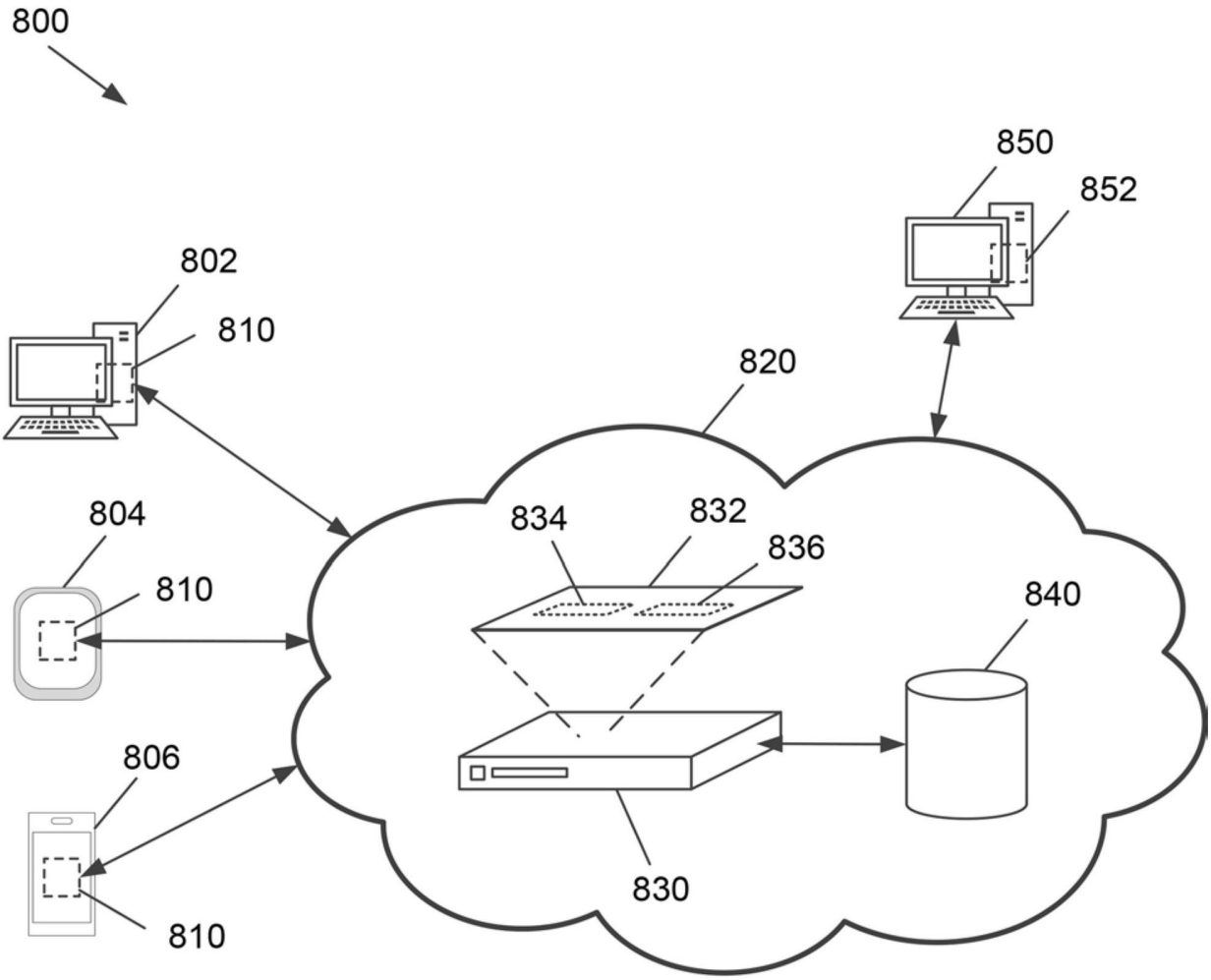


图8

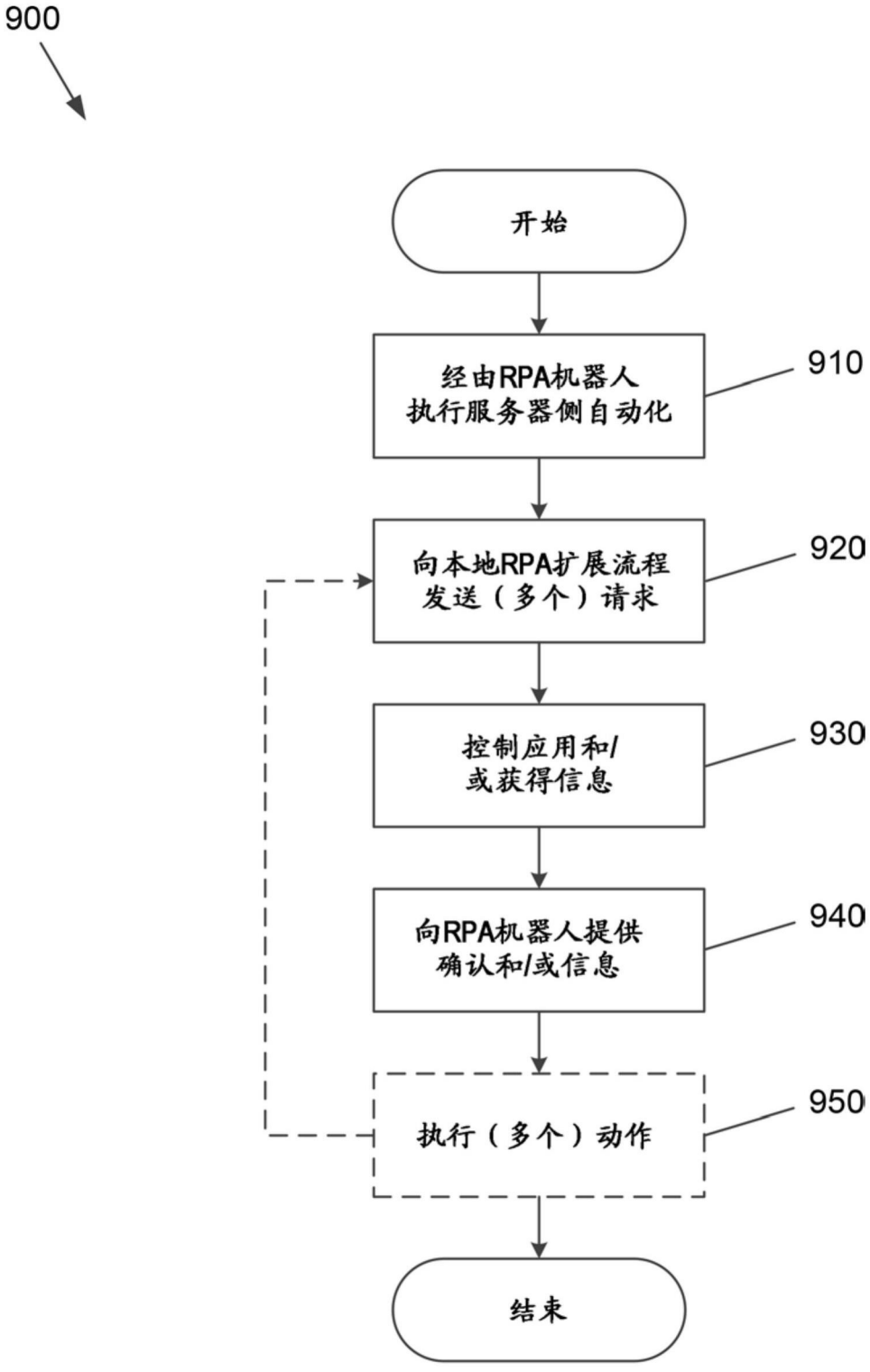

900
开始
经由RPA机器人
执行服务器侧自动化
910
向本地RPA扩展流程
发送（多个）请求
920
控制应用和/
或获得信息
930
向RPA机器人提供
确认和/或信息
940
执行（多个）动作
950
结束

图9

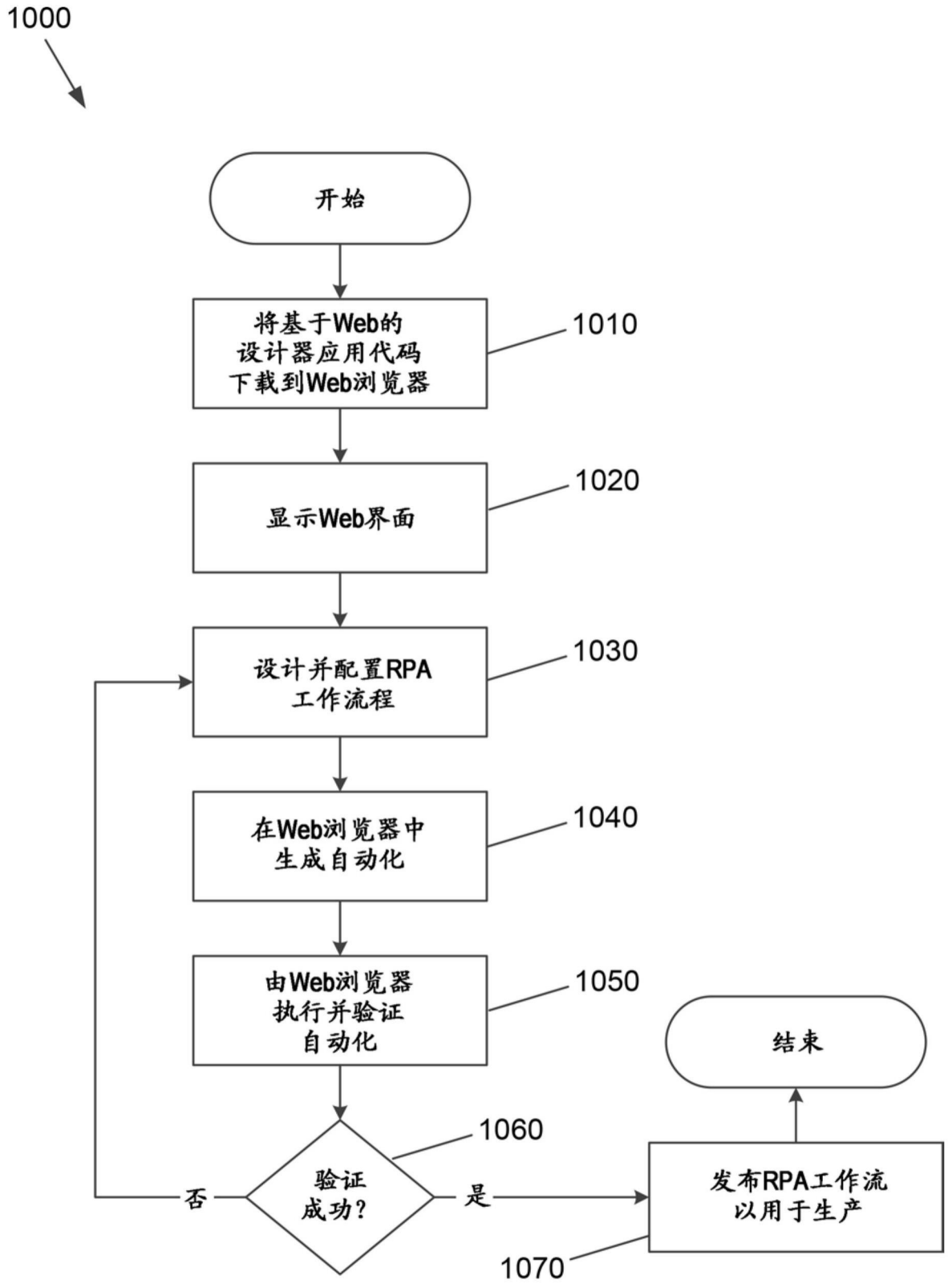

1000
开始
将基于Web的
设计器应用代码
下载到Web浏览器
1010
显示Web界面
1020
设计并配置RPA
工作流程
1030
在Web浏览器中
生成自动化
1040
由Web浏览器
执行并验证
自动化
1050
验证
成功?
1060
否
是
发布RPA工作流
以用于生产
1070
结束

图10

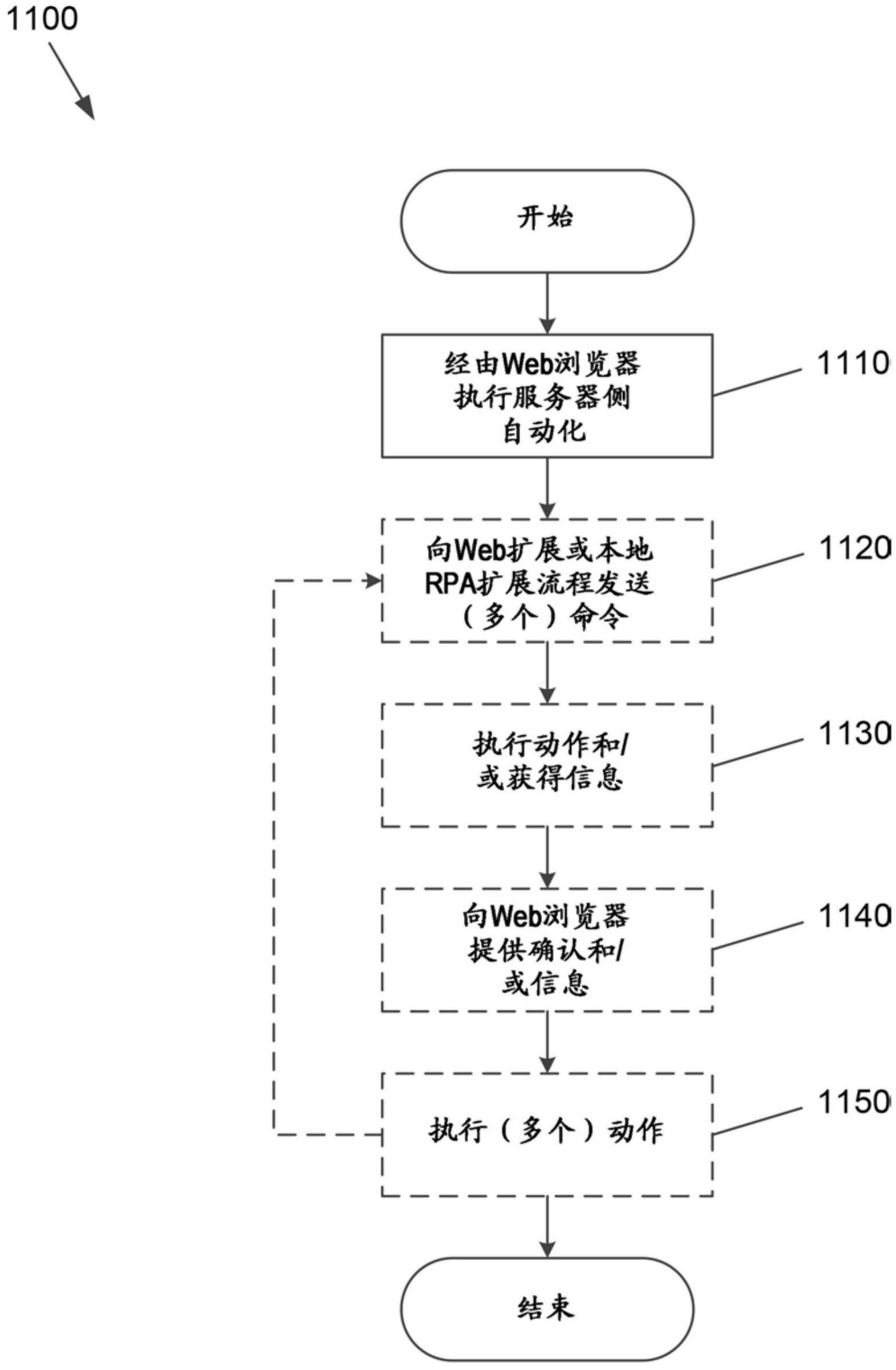

1100
开始
经由Web浏览器
执行服务器侧
自动化
1110
向Web扩展或本地
RPA扩展流程发送
（多个）命令
1120
执行动作和/
或获得信息
1130
向Web浏览器
提供确认和/
或信息
1140
执行（多个）动作
1150
结束

图11