



(12) 发明专利

(10) 授权公告号 CN 116057504 B

(45) 授权公告日 2025.05.16

(21) 申请号 202080104226.X

(74) 专利代理机构 北京市金杜律师事务所

(22) 申请日 2020.09.24

11256

(65) 同一申请的已公布的文献号

专利代理人 李兴斌

申请公布号 CN 116057504 A

(51) Int.CI.

G06F 9/451 (2006.01)

(43) 申请公布日 2023.05.02

G06F 3/0484 (2006.01)

(30) 优先权数据

G06F 3/0481 (2006.01)

16/922,289 2020.07.07 US

G06Q 10/06 (2006.01)

(85) PCT国际申请进入国家阶段日

G06Q 10/10 (2006.01)

2023.02.07

G06F 8/38 (2006.01)

(86) PCT国际申请的申请数据

G06F 8/36 (2006.01)

PCT/US2020/052429 2020.09.24

G06F 8/34 (2006.01)

(87) PCT国际申请的公布数据

(56) 对比文件

W02022/010516 EN 2022.01.13

CN 104423969 A, 2015.03.18

(73) 专利权人 尤帕斯公司

EP 3112965 A1, 2017.01.04

地址 美国纽约州

审查员 姚翠

(72) 发明人 M·格里戈尔

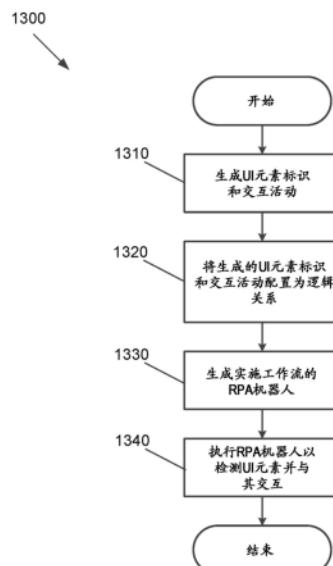
权利要求书4页 说明书22页 附图62页

(54) 发明名称

创建机器人流程自动化的用户界面的计算机程序产品及方法

(57) 摘要

公开了用于机器人流程自动化(RPA)的用户界面(UI)对象描述符、UI对象库、UI对象储存库和UI对象浏览器。UI对象浏览器可用于管理、重用和增加项目中UI描述符的可靠性。UI描述符可被添加到UI对象库中，并且作为UI对象库被发布或重新发布，以用于UI对象储存库中的全局重用。UI对象浏览器、UI对象库和UI对象储存库可促进UI元素标识框架及其衍生物的可重用性。



1. 一种用于创建针对机器人流程自动化RPA的用户界面UI对象库的计算机程序产品，所述计算机程序产品包括计算机程序，所述计算机程序被配置为使至少一个处理器：

检测与UI的屏幕相关联的应用；

检测所述屏幕上的UI元素；

接收对目标UI元素的选择；

生成针对所选择的所述UI元素的UI描述符，所生成的所述UI描述符包括用于在所述屏幕上查找所述目标UI元素的指令集合；

将针对所选择的所述UI元素的所生成的所述UI描述符添加到相应的RPA工作流活动；以及

在UI对象储存库中创建和发布包括所述UI描述符的UI对象库，

其中所述UI对象库包括与来自检测到的所述应用的版本的一个或多个屏幕相对应的UI描述符。

2. 根据权利要求1所述的计算机程序产品，其中所述UI描述符是或者包括使用驱动程序基于最佳匹配生成的选择符。

3. 根据权利要求1所述的计算机程序产品，其中所述RPA工作流活动是或者包括基于选择符的标识活动、基于计算机视觉CV的标识活动、统一目标标识活动、图像匹配标识活动或者基于选择符和光学字符识别OCR的标识活动。

4. 根据权利要求1所述的计算机程序产品，其中

与所述UI的所述屏幕相关联的所述应用的所述检测包括自动检测所述应用和所述应用的版本，

针对所述RPA工作流活动，检测到的所述应用和所述应用的所述版本被添加，并且

所生成的所述UI描述符是针对检测到的所述应用的版本的UI描述符。

5. 根据权利要求1所述的计算机程序产品，其中所述计算机程序还被配置为使所述至少一个处理器：

从包括所述RPA工作流活动的RPA工作流中的活动提取UI描述符；以及

将所提取的所述UI描述符添加到按UI应用、版本、屏幕和UI元素对所述UI描述符进行分组的结构化模式。

6. 根据权利要求5所述的计算机程序产品，其中所述结构化模式被布置在本体中，所述本体包括不同屏幕中的UI元素之间的关系。

7. 根据权利要求1所述的计算机程序产品，其中检测到的所述应用是web应用，并且所述UI描述符包括到针对对应选择符的相应UI元素的超文本标记语言HTML路径。

8. 根据权利要求1所述的计算机程序产品，其中所述计算机程序是或者包括RPA开发者应用。

9. 根据权利要求1所述的计算机程序产品，其中所述计算机程序还被配置为使所述至少一个处理器：

从UI对象储存库检索一个或多个其他对象库；以及

显示所创建的所述UI对象库和所述一个或多个其他对象库。

10. 根据权利要求1所述的计算机程序产品，其中所述计算机程序是或者包括UI对象浏览器，所述UI对象浏览器显示UI对象储存库的一个或多个UI对象库中的UI描述符，所述一

个或多个UI对象库按应用和版本被组织。

11. 根据权利要求1所述的计算机程序产品,其中所述RPA工作流活动是或者包括基于计算机视觉CV的活动,并且所述计算机程序还被配置为使所述至少一个处理器:

在所述UI的所述屏幕上执行CV以标识所述屏幕上的UI元素;

显示具有由所述CV标识的可选择组件的所述UI的所述屏幕;

接收对一个或多个锚点UI元素的选择,以帮助唯一地标识所述目标UI元素;以及

将所述一个或多个锚点添加到所述RPA工作流活动,以辅助标识所述目标UI元素。

12. 根据权利要求1所述的计算机程序产品,其中所述RPA工作流活动是或者包括基于统一目标的活动,并且在所述目标UI元素不能被唯一地标识时,所述计算机程序还被配置为使所述至少一个处理器:

接收对一个或多个锚点UI元素的选择,以帮助唯一地标识所述目标UI元素;

验证所选择的所述一个或多个锚点UI元素唯一地标识至少具有最小验证得分的所述目标UI元素;以及

在所选择的所述一个或多个锚点UI元素唯一地标识至少具有所述最小验证得分的所述目标UI元素时,

将所述一个或多个锚点添加到所述RPA工作流活动,以辅助标识所述目标UI元素。

13. 根据权利要求1所述的计算机程序产品,其中所述RPA工作流活动是或者包括基于图像匹配的活动,所述基于图像匹配的活动使用图像匹配技术搜索由用户指定的在所述屏幕上的图像。

14. 根据权利要求1所述的计算机程序产品,其中所述RPA工作流活动是或者包括基于选择符和光学字符识别OCR的活动,所述基于选择符和OCR的活动使用选择符和与所述目标UI元素相关联的文本的OCR的组合来标识所述目标UI元素,所述选择符是使用驱动程序基于最佳匹配而被生成的。

15. 根据权利要求1所述的计算机程序产品,其中所述计算机程序还被配置为使所述至少一个处理器:

将包括所述RPA工作流活动的多个RPA工作流活动配置成逻辑关系。

16. 根据权利要求1所述的计算机程序产品,其中所述计算机程序还被配置为使所述至少一个处理器:

生成实现包括所述RPA工作流活动的RPA工作流的RPA机器人。

17. 根据权利要求1所述的计算机程序产品,其中所生成的所述UI描述符是针对检测到的所述应用的新版本的UI描述符。

18. 一种计算机程序产品,所述计算机程序产品包括计算机程序,所述计算机程序被配置为使至少一个处理器:

接收对应用的屏幕上的目标用户界面UI元素的选择;

生成针对所选择的所述UI元素的UI描述符,所生成的所述UI描述符包括用于在所述屏幕上查找所述目标UI元素的指令集合;以及

将所述生成的针对所选择的所述UI元素的所述UI描述符添加到相应的机器人流程自动化RPA工作流活动,其中

所述RPA工作流活动是或者包括基于计算机视觉CV的标识活动或者统一目标标识活

动。

19. 根据权利要求18所述的计算机程序产品,其中所述计算机程序还被配置为使所述至少一个处理器:

从包括所述RPA工作流活动的RPA工作流中的活动提取UI描述符;以及

将所提取的所述UI描述符添加到按UI应用、版本、屏幕和UI元素对所述UI描述符进行分组的结构化模式。

20. 根据权利要求19所述的计算机程序产品,其中所述结构化模式被布置在本体中,所述本体包括不同屏幕中的UI元素之间的关系。

21. 根据权利要求18所述的计算机程序产品,其中所述计算机程序还被配置为使所述至少一个处理器:

在UI对象储存库中创建和发布包括所述UI描述符的UI对象库,

其中所述UI对象库包括与来自检测到的所述应用的版本的一个或多个屏幕相对应的UI描述符。

22. 根据权利要求18所述的计算机程序产品,其中所述计算机程序是或者包括UI对象浏览器,所述UI对象浏览器显示UI对象储存库的一个或多个UI对象库中的UI描述符,所述一个或多个UI对象库按应用和版本被组织。

23. 根据权利要求18所述的计算机程序产品,其中所述RPA工作流活动是或者包括所述基于CV的标识活动,并且所述计算机程序还被配置为使所述至少一个处理器:

在所述UI的所述屏幕上执行CV以标识所述屏幕上的UI元素;

显示具有由所述CV标识的可选择组件的所述UI的所述屏幕;

接收对一个或多个锚点UI元素的选择,以帮助唯一地标识所述目标UI元素;以及

将所述一个或多个锚点添加到所述RPA工作流活动,以辅助标识所述目标UI元素。

24. 根据权利要求18所述的计算机程序产品,其中所述RPA工作流活动是或者包括基于统一目标的活动,并且在所述目标UI元素不能被唯一地标识时,所述计算机程序还被配置为使所述至少一个处理器:

接收对一个或多个锚点UI元素的选择,以帮助唯一地标识所述目标UI元素;

验证所选择的所述一个或多个锚点UI元素唯一地标识至少具有最小验证得分的所述目标UI元素;以及

在所选择的所述一个或多个锚点UI元素唯一地标识至少具有最小验证得分的所述目标UI元素时,

将所述一个或多个锚点添加到所述RPA工作流活动,以辅助标识所述目标UI元素。

25. 一种机器人流程自动化RPA的方法,应用于RPA开发者应用,所述方法包括:

检测与用户界面UI的屏幕相关联的应用;

检测所述屏幕上的UI元素;

接收对目标UI元素的选择;

生成针对所选择的所述UI元素的UI描述符,所生成的所述UI描述符包括用于在所述屏幕上查找所述目标UI元素的指令集合;以及

将所生成的针对所选择的所述UI元素的所述UI描述符添加到相应的RPA工作流活动,其中

所述RPA工作流活动是或者包括基于选择符的标识活动、基于计算机视觉CV的标识活动、统一目标标识活动、图像匹配标识活动或者基于选择符和光学字符识别OCR的标识活动,以及

所述RPA开发者应用包括UI对象浏览器,所述UI对象浏览器显示UI对象储存库的一个或多个UI对象库中的UI描述符,所述一个或多个UI对象库按应用和版本被组织。

26.根据权利要求25所述的方法,还包括:

在所述UI对象储存库中创建和发布包括所述UI描述符的UI对象库,其中所生成的所述UI描述符是针对检测到的所述应用的新版本的UI描述符。

27.根据权利要求25所述的方法,还包括:

从包括所述RPA工作流活动的RPA工作流中的活动提取UI描述符;以及

将所提取的所述UI描述符添加到按UI应用、版本、屏幕和UI元素对所述UI描述符进行分组的结构化模式,其中

所述结构化模式被布置在本体中,所述本体包括不同屏幕中的UI元素之间的关系。

创建机器人流程自动化的用户界面的计算机程序产品及方法

[0001] 相关申请的交叉引用

[0002] 本申请要求于2020年7月7日提交的美国非临时专利申请第16/922,289号的权益。该先前提交的申请的主题在此通过引用整体并入本文。

技术领域

[0003] 本发明总体上涉及机器人流程自动化(RPA),并且更具体地,涉及用于RPA的用户界面(UI)对象描述符、UI对象库、UI对象储存库和/或UI对象浏览器。

背景技术

[0004] 对于用户界面(UI)中的RPA自动化,开发者传统地为每个UI动作构建描述符(例如,选择符)。传统地会生成默认描述符。然而,构建可靠的描述符可能会消耗开发者用于故障排除和试错开发的时间,这可能效率不高。当需要相同的描述符时,也缺乏描述符的可重用性。此外,当系统和应用改变或升级时,流程升级路径可能并不简单。因此,改进的方法可能是有益的。

发明内容

[0005] 本发明的某些实施例可以是当前RPA技术尚未完全标识、理解或解决的现有技术中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用于RPA的UI描述符、UI对象库、UI对象储存库和/或UI对象浏览器。

[0006] 在实施例中,用于为RPA创建用户界面(UI)对象库的计算机程序体现在非暂态计算机可读介质上。该程序被配置为使至少一个处理器检测与UI的屏幕相关联的应用,并且检测屏幕上的UI元素。该程序还被配置为使至少一个处理器接收对目标UI元素的选择,并且为所选择的UI元素生成UI描述符,并且将所选择的UI元素的所生成的UI描述符添加到相应的RPA工作流活动。该程序进一步被配置为使至少一个处理器在UI对象储存库中创建和发布包括UI描述符的UI对象库。

[0007] 在另一实施例中,计算机程序体现在非暂态计算机可读介质上。该程序被配置为使至少一个处理器接收对应用的屏幕上的目标UI元素的选择,并且为所选择的UI元素生成UI描述符。该程序还被配置为使至少一个处理器将所选择的UI元素的所生成的UI描述符添加到相应的RPA工作流活动。RPA工作流活动是或包括基于CV的标识活动或统一目标标识活动。

[0008] 在另一实施例中,RPA开发者应用体现在非暂态计算机可读介质上。RPA开发者应用被配置为使至少一个处理器检测与UI的屏幕相关联的应用,并且检测屏幕上的UI元素。RPA应用还被配置为使至少一个处理器接收对目标UI元素的选择,并且为所选择的UI元素生成UI描述符。RPA开发者应用还被配置为使至少一个处理器将所选择的UI元素的所生成的UI描述符添加到相应的RPA工作流活动。RPA工作流活动是或包括基于选择符的标识活动、基于CV的标识活动、统一目标标识活动、图像匹配标识活动或基于选择符和OCR的标识

活动。RPA应用包括显示UI对象储存库的一个或多个UI对象库中的UI描述符的UI对象浏览器，该一个或多个UI对象库由应用和版本组织。

附图说明

[0009] 为了便于理解本发明的某些实施例的优点，将通过参考附图中所示的特定实施例来呈现上面简要描述的本发明的更详细的描述。尽管应理解，这些附图仅描绘了本发明的传统实施例，因此不应被认为是对其范围的限制，但将通过使用附图以附加的特征和细节对本发明进行描述和解释，在附图中：

- [0010] 图1是图示根据本发明的实施例的机器人流程自动化(RPA)系统的架构图。
- [0011] 图2是图示根据本发明的实施例的已部署RPA系统的架构图。
- [0012] 图3是图示根据本发明的实施例的设计器、活动和驱动程序之间关系的架构图。
- [0013] 图4是图示根据本发明的实施例的RPA系统的架构图。
- [0014] 图5是图示根据本发明的实施例的被配置为提供用于RPA的UI描述符、对象库、UI对象储存库和/或对象浏览器的计算系统的架构图。
- [0015] 图6A是图示根据本发明的实施例的RPA设计器应用的屏幕截图。
- [0016] 图6B是图示Windows[®] 7计算器标准视图屏幕和视图选择符选项卡的屏幕截图。
- [0017] 图6C是图示根据本发明的实施例的如图6A所示的RPA开发应用的大纲窗格中的序列的放大屏幕截图视图。
- [0018] 图6D是图示根据本发明的实施例的如图6A所示的RPA开发应用的摘录窗格的放大屏幕截图视图。
- [0019] 图6E和6F是图示根据本发明的实施例的RPA工作流的活动的屏幕截图，每个活动具有相应的UI元素的屏幕截图。
- [0020] 图6G是图示根据本发明的实施例的工作流活动选项菜单的屏幕截图。
- [0021] 图6H是图示根据本发明的实施例的具有属性窗格中示出的点击活动的属性的RPA设计器应用的屏幕截图。
- [0022] 图6I是图示根据本发明的实施例的如图6H所示的RPA开发应用的属性窗格的放大屏幕截图。
- [0023] 图6J是图示根据本发明的实施例的具有UI对象浏览器窗格的RPA设计器应用的屏幕截图。
- [0024] 图6K是图示根据本发明的实施例的如图6J所示的RPA开发应用的UI对象浏览器窗格的UI描述符视图的放大屏幕截图。
- [0025] 图6L是图示根据本发明的实施例的如图6J所示的RPA开发应用的UI对象浏览器窗格的UI描述符视图的放大屏幕截图。
- [0026] 图6M是图示根据本发明的实施例的UI对象浏览器窗格的UI活动视图的屏幕截图。
- [0027] 图6N是图示根据本发明的实施例的用于将选择符放入RPA工作流的期望活动中的拖放功能性的屏幕截图。
- [0028] 图6O是图示Windows[®] 10计算器的屏幕截图。

- [0029] 图6P是图示根据本发明的实施例的具有Windows[®] 10计算器选择符的UI对象浏览器窗格的UI活动视图的屏幕截图。
- [0030] 图6Q是图示根据本发明的实施例的RPA工作流的活动中的一些的屏幕截图，其中具有用于Windows[®] 10计算器的每个活动的相应UI元素的更新屏幕截图。
- [0031] 图6R是图示根据本发明的实施例的发布UI对象库窗口的屏幕截图。
- [0032] 图6S是图示根据本发明的实施例的管理包窗口的屏幕截图。
- [0033] 图6T是图示根据本发明的实施例的UI对象浏览器窗格的UI活动视图的屏幕截图，其中将Windows[®] 10计算器选择符添加到UI对象库中。
- [0034] 图7A是图示根据本发明的实施例的RPA设计器应用的屏幕截图。
- [0035] 图7B是图示根据本发明的实施例的将被配置为使用三种不同UI元素检测技术的RPA工作流的屏幕截图。
- [0036] 图7C是图示根据本发明的实施例，在开发者选择“在屏幕上指示”之后，由UiPath StudioTM修改的Windows[®] 10计算器的屏幕截图。
- [0037] 图7D是图示根据本发明的实施例的具有来自Windows[®] 10计算器的所选择的UI元素的屏幕截图的图7B的RPA工作流的屏幕截图。
- [0038] 图7E是图示根据本发明的实施例的活动选项菜单的屏幕截图。
- [0039] 图7F是图示根据本发明的实施例的选择符编辑器的屏幕截图。
- [0040] 图7G是图示根据本发明的实施例的添加到UI对象库功能性的屏幕截图。
- [0041] 图7H是图示根据本发明的实施例的UI描述符窗格的屏幕截图。
- [0042] 图7I是图示根据本发明的实施例的具有检测屏幕元素和标签消息的RPA工作流的屏幕截图。
- [0043] 图7J是图示根据本发明的实施例的搜索界面的屏幕截图。
- [0044] 图7K是图示根据本发明的实施例的基于CV的标识活动的屏幕截图。
- [0045] 图7L是图示根据本发明的实施例，在开发者通过CV点击界面选择“在范围上指示”之后，由UiPath StudioTM修改的Windows[®] 10计算器的屏幕截图。
- [0046] 图7M是图示根据本发明的实施例，由UiPath StudioTM修改的Windows[®] 10计算器和在开发者进行目标选择之后的CV点击界面的屏幕截图。
- [0047] 图7N是图示根据本发明的实施例，由UiPath StudioTM修改的Windows[®] 10计算器和在开发者进行目标选择和锚点选择之后的CV点击界面的屏幕截图。
- [0048] 图7O是图示根据本发明的实施例，在利用目标和锚点配置基于CV的标识活动之后，工作流中基于CV的标识活动的屏幕截图。
- [0049] 图7P是图示根据本发明的实施例的具有带有描述符属性的CV点击的属性选项卡的屏幕截图。
- [0050] 图7Q是图示根据本发明的实施例的基于统一目标的标识活动的屏幕截图。
- [0051] 图7R是图示根据本发明的实施例的具有突出显示的统一目标点击活动的搜索界面的屏幕截图。

[0052] 图7S是图示根据本发明的实施例的具有嵌套点击活动的基于统一目标的标识活动的屏幕截图。

[0053] 图7T是图示根据本发明的实施例的由UiPath StudioTM修改的Windows[®] 10计算器和统一目标选择选项窗口的屏幕截图。

[0054] 图7U是图示根据本发明的实施例的具有所选目标的由UiPath StudioTM修改的Windows[®] 10计算器以及统一目标选择选项窗口的屏幕截图。

[0055] 图7V是图示根据本发明的实施例的具有所选目标和所选锚点的由UiPath StudioTM修改的Windows[®] 10计算器以及统一目标选择选项窗口的屏幕截图。

[0056] 图7W是图示根据本发明的实施例的由UiPath StudioTM修改的Windows[®] 10计算器的屏幕截图，其中所选目标和所选锚点经由统一目标选择选项窗口进行验证。

[0057] 图7X是图示根据本发明的实施例的具有嵌套点击活动中的目标和锚点的屏幕截图的基于统一目标的标识活动的屏幕截图。

[0058] 图7Y是图示根据本发明的实施例的添加到工作流的点击图像活动的屏幕截图。

[0059] 图7Z图示了根据本发明的实施例，由UiPath StudioTM修改的Windows[®] 10计算器的屏幕截图，其中在待点击的图像周围绘制了一个框，并且在点击图像活动中从选择框中选择了该图像的屏幕截图。

[0060] 图7a是图示根据本发明的实施例的具有点击图像活动属性的属性选项卡的屏幕截图。

[0061] 图7b是图示根据本发明的实施例的待点击图像的选择符的选择符编辑器的屏幕截图。

[0062] 图7c是图示根据本发明的实施例的工作流中点击OCR文本活动的屏幕截图。

[0063] 图7d是图示根据本发明的实施例的具有待点击按钮的屏幕截图的点击OCR文本活动的屏幕截图。

[0064] 图8是图示根据本发明的实施例的用于生成基于选择符的活动的流程的流程图。

[0065] 图9是图示根据本发明的实施例的用于生成基于CV描述符的活动的流程的流程图。

[0066] 图10是图示根据本发明的实施例的用于生成基于统一目标的活动的流程的流程图。

[0067] 图11是图示根据本发明的实施例的用于生成图像匹配活动的流程的流程图。

[0068] 图12是图示根据本发明的实施例的用于生成基于选择符和OCR的活动的流程的流程图。

[0069] 图13是图示根据本发明的实施例的用于生成使用多种标识技术的RPA工作流的流程的流程图。

[0070] 图14是图示根据本发明的实施例的用于创建UI对象库和配置项目以与新版本的应用一起工作的流程的流程图。

[0071] 除非另有说明，否则在整个附图中，相似的附图标记始终表示对应的特征。

具体实施方式

[0072] 一些实施例涉及用于RPA的UI描述符、UI对象库、UI对象储存库和/或UI对象浏览器。屏幕上的UI元素(例如,文本字段、按钮、标签、菜单、复选框等)可按应用、应用的版本、应用屏幕和UI元素的集合进行分组,并且每个屏幕具有多个UI元素。如本文所使用,“屏幕”是应用UI或应用UI的一部分在某个时间点的图像。在此上下文中,“应用”或给定应用的版本可以是屏幕的联合。在一些实施例中,每个UI元素可由一个或多个UI描述符来描述。UI元素、UI描述符、应用和应用屏幕均是UI对象。在一些实施例中,UI元素和屏幕可进一步区分成特定类型的UI元素(例如,按钮、复选框、文本字段等)以及屏幕(例如,顶部窗口、模态窗口、弹出窗口等)。

[0073] UI对象浏览器可访问可按应用、应用的版本、应用屏幕、UI元素的集合及其组合等进行分组的UI对象库。在一些实施例中,UI对象库可存储在UI对象储存库中。如本文所使用,UI对象储存库是UI对象库的集合。在一些实施例中,UI对象储存库可以是NuGetTM订阅源、web服务等。

[0074] 对象浏览器可在一些实施例中用于导航、管理和编辑UI对象储存库中的UI对象库。在一些实施例中,UI对象库的UI对象储存库可促进管理、重用和增加项目中UI描述符的可靠性。在一些实施例中,UI描述符可被添加到UI库中,并且被发布或重新发布以供UI对象储存库中的全局重用。UI对象浏览器通过提供对UI对象储存库及其UI对象库的访问,可促进UI元素标识框架及其衍生物的可重用性。

[0075] 为使UI对象可重用,可将其提取到可由RPA流程引用的UI对象库中。例如,当选择符或其他UI描述符由于应用的新版本而被修改时,库可被重新创建(或重新发布)以包括经修改的UI描述符。然后,使用UI对象库的RPA流程可调用经修改的UI描述符版本。在一些实施例中,对于RPA流程,对与新库相关联的新应用编程界面(API)的引用可由开发者手动修改、经由软件工具(例如,当选择符或其他组件改变时,经历RPA流程并更新UI描述符引用的工具)修改,或者自动修改。

[0076] 在一些实施例中,UI对象的包可作为依赖项被附加(例如,NuGetTM包)。然而,在某些实施例中,UI对象可经由对web服务的一个或多个API调用来获得。这使得UI对象能够被远程存储,并且在自动化期间被检索和使用。

[0077] UI描述符的初始构建先前要求开发者对UI描述符背后发生的事情具有技术知识。然而,许多RPA流程不再由软件开发者构建。例如,这些人可使用UiPath StudioXTM创建RPA工作流,这不需要深入的编程知识。一些实施例允许创建RPA工作流的个人重用描述符,这提供了时间节约,因为如果合适的UI描述符对创建RPA工作流的个人可用,则不需要软件开发者从头开始构建新的UI描述符。

[0078] 添加到UI对象库的UI元素可表示用于RPA工作流的选择符。为了自动化UI中的特定操作,RPA机器人可与各种窗口、按钮、下拉列表和/或其他图形元素进行交互。按照惯例,这是使用UI元素的预期屏幕位置来完成的。然而,这是不可靠的。

[0079] 一些实施例使用选择符,通过将UI元素及其父元素的特性存储在XML片段中,克服了与固定坐标标识相关联的问题。尽管在UI是静态的一些实施例中可自动生成选择符,但一些软件程序,诸如一些web应用,具有可变值的变化的布局和特性节点。这些变化可能不容易预测,并且这先前需要手动生成一些选择符。然而,一些实施例的对象浏览器可克服这

一问题。

[0080] 在一些实施例中,选择符是可用于检测UI元素的UI描述符的类型。在一些实施例中,选择符具有以下结构:

[0081] <node_1/><node_2/>...<node_N/>

[0082] 最后的节点表示感兴趣的GUI元素,并且所有先前的节点表示该元素的父元素。<node_1>通常被称为根节点,并且表示应用的顶部窗口。

[0083] 每个节点可具有有助于正确标识所选择的应用的特定级别的一种或多种特性。在一些实施例中,每个节点具有以下格式:

[0084] <ui_system attr_name_1='attr_value_1'...attr_name_N='attr_value_N' />

[0085] 每个特性可具有分配的值,并且可选择具有常量值的特性。这是因为每次启动应用时对特性值的更改可能会使得选择符无法正确标识相关联的元素。

[0086] UI对象库描述符可直接添加到RPA工作流活动中,从而节省了开发者为活动创建自定义选择符所需的时间。对象浏览器可提供在对象库中存储创建的选择符的数据库,以实现UI描述符的可重用性。对象库在本文中被定义为对应于来自应用的某个版本的一个或多个屏幕的UI描述符的集合。UI描述符是用于查找UI元素的指令集合。在一些实施例中,UI描述符是封装的数据/结构格式,其包括(多个)UI元素选择符、(多个)锚点选择符、(多个)计算机视觉(CV)描述符、(多个)统一目标描述符、(多个)屏幕图像捕获(上下文)、元素图像捕获、其他元数据(例如,应用和应用版本)、其组合等。封装的数据/结构格式可随着平台的未来更新而扩展,并且不限于上述定义。在不偏离本发明的范围的情况下,可使用用于标识屏幕上的UI元素的任何合适的UI描述符。统一目标描述符将多种类型的UI描述符链接在一起。统一目标描述符可像有限状态机(FSM)一样起作用,其中在第一上下文中,应用第一UI描述符机制,在第二上下文中,应用第二UI描述符,等等。

[0087] 在一些实施例中,RPA设计器应用可向用户询问用户计划自动化何种(多个)类型的应用。例如,用户可指定Excel[®]、Workday[®]、SAP[®]等。由于RPA应用已经包括用于那些应用的UI对象库中的屏幕描述符,因此可利用关于如何自动化这些应用的逻辑来对应用进行编程。这些应用的各种版本的开箱即用选择符可能是可用的,并且用户可能能够指定要自动化的(多个)版本。

[0088] 如本文所使用,术语“用户”和“开发者”可互换使用。用户/开发者可能具有或不具有编程和/或技术知识。例如,在一些实施例中,用户/开发者可通过配置RPA工作流中的活动来创建RPA工作流,而无需手动编码。在某些实施例中,这可通过例如点击和拖放各种特征来完成。

[0089] 按照惯例,使选择符可靠需要测试、了解选择符的内在细节等。这使得没有技术知识的人难以或不可能创建/改进选择符。通过为公共UI元素和任务提供正常运行的选择符的库,一些实施例允许非技术人员开发在生产中工作的自动化。用户可指示要自动化的应用版本的屏幕,并且RPA开发应用可获取该屏幕。然后,用户可与屏幕交互来定义他或她想要做什么。

[0090] 在一些实施例中,随着创建新的UI描述符和/或修改现有的UI描述符,可建立UI对象库的全局数据库,该全局数据库可共享、协作且可能开源。在某些实施例中,可使用分类

法和本体。应用、版本、屏幕、UI元素、描述符等可被定义为一个分类法，这是子类别的分层结构。

[0091] 然而，许多现实世界的概念不容易进行分类组织。相反，它们可能更接近于数学本体的概念。在本体中，类别之间的关系不一定是分层的。例如，屏幕上的按钮在被点击时将用户带到另一个屏幕的情况不能被该屏幕的分类容易地捕获，因为下一个屏幕不在分层结构中。当构建表示这种情况的图形时，应用的对象浏览器可被建模为本体，该本体允许在同一屏幕或不同屏幕上的UI元素之间创建交互，并且提供关于UI元素如何彼此相关的更多信息。

[0092] 考虑点击确定按钮进入员工屏幕的示例。本体的结构可使设计器应用建议用户在下一屏幕上筛选员工。经由确定按钮，这些屏幕之间的关系的本体信息允许设计器应用这样做。通过定义不一定是树的图形结构，而是与应用实际正在做的事情相关的图形结构，可捕获更复杂和丰富的关系。

[0093] 一些实施例属于或包括允许更快地从应用中获取屏幕和UI元素的记录器。例如，如果给定的屏幕有250个不同的UI元素，则可能需要一些时间才能逐个获取它们。屏幕指针可用于获取所有可编辑的UI元素。这可能有助于创建树或本体的第一版本。

[0094] 在一些实施例中，对象浏览器可被暴露为web服务。如果用户在运行时更新了选择符或其他UI描述符，如果进行了适当配置，RPA机器人可能会调用最新版本的服务。这允许RPA机器人始终使用当前版本的UI描述符。

[0095] 可从RPA工作流中的活动中提取UI描述符，并且将其添加到按UI应用、屏幕和UI元素对UI描述符进行分组的结构化模式中。在一些实施例中，UI描述符可以是用于广泛重用的一个项目的一部分、用于测试目的的全局储存库的一部分、或者用于全局跨项目共享的UI对象库的一部分。在一些实施例中，对象库是通过应用、应用的版本和屏幕进行分组的UI描述符的封装。可以定义UI描述符并将其添加到对象库中，在一些实施例中，该对象库可在发布之后作为依赖项安装在其他项目中。这使得UI描述符的对象库可供重用。在一些实施例中，UI应用是具有多个版本的目标应用，其中每个版本具有多个屏幕。

[0096] 出于各种原因，UI对象储存库和UI描述符重用可能是有益的。例如，如果应用中的UI元素的位置、外观和/或功能性改变，则UI描述符可被改变，并且该改变然后可被传播到使用UI描述符的活动。因此，可在标识UI中的图形元素的级别上提供可重用性。

[0097] 这对于适应新版本的应用可能特别有利。对于新版本，软件开发者可能只需要更新UI描述符和/或从头开始开发一些新的选择符，从而大大减少了开发时间。例如，对于网页，可检查超文本标记语言 (HTML) 以获得针对对应的选择符的UI元素的路径。对应的UI元素的选择符在不同的web浏览器和/或同一web浏览器的不同版本中可能不同。这个概念可能也适用于可视桌面、服务器、智能手机和平板计算机应用。UI对象库可能包括应用的实际选择符，根据上面的描述，这只是一种类型的UI描述符。对于一个或多个应用，UI对象库可具有多个不同类型的UI描述符。UI描述符也可指构建应用的不同技术。例如，不同的UI描述符可被创建并用于独立的桌面、web和移动应用。对于一种技术，您可能希望使用选择符，而对于另一种技术，您可能希望使用CV描述符，等等。

[0098] 在一些实施例中，通过在涵盖所有UI检测机制的统一目标中执行图像检测和定义，UI描述符可与统一目标一起工作。统一目标可将标识和自动化UI元素的多种技术合并

到单一的内聚方法中。统一目标可优先考虑基于选择符和基于驱动程序的UI检测机制，并且如果前两种机制不成功，则后退到CV以找到图像。在一些实施例中，选择符编辑器和UI浏览器可被配置为支持统一目标。

[0099] 在一些实施例中，设计器应用包括作为从一个或多个源加载对象库的“对象浏览器”的部分或面板。该对象浏览器可允许用户可视化整个UI对象集合，并且经由拖放和向导或上下文动作来使用UI描述符中的任何一个。这可能允许用户选择要在RPA工作流活动中执行的操作，诸如点击、获取文本、键入等。

[0100] 下文提供了对象库结构的示例。

- [0101] • SAP
- [0102] • 版本1
- [0103] • 屏幕1
- [0104] • 对象1
- [0105] • 对象2
- [0106] • ...
- [0107] • 屏幕2...
- [0108] • ...
- [0109] • 版本2...
- [0110] • ...
- [0111] • Salesforce...
- [0112] • ...

[0113] 应注意，上述分层结构仅作为示例提供。在不偏离本发明的范围的情况下，可使用分层结构中任何期望数目的级别以及与每个级别相关联的元素。在某些实施例中，用户可根据他或她的需要来定义应用地图。此外，在一些实施例中，树中的一些节点可能仅充当管理角色，而不是功能角色。例如，UI元素可能被组合到一个没有UI描述符的容器中。在一些实施例中，容器可仅出于分组的目的而存在。

[0114] 在一些实施例中，用户可通过从对象库中可用的屏幕集合中进行选择来在设计器应用中限定范围。当范围被一对一地绑定到某个屏幕时，用户可添加UI动作（例如，点击、获取文本等），然后使用对潜在候选选项的智能感知来将这些动作映射到UI对象。在一些实施例中，对象集合可限于针对UI对象储存库（例如，UI对象库的数据库）内的屏幕定义的列表。

[0115] 例如，一旦确定了应用屏幕、UI元素等的结构，可提供标识的应用的列表。在某些实施例中，用户然后可选择用于自动化的应用，将所选择的应用拖到画布上，写入“SAP”，按下“.”，并且可列出所有SAP[®]屏幕。选择屏幕之后，可出现附加候选元素、功能等。这允许用户选择预测的UI元素、功能、应用和屏幕。

[0116] 在一些实施例中，例如，当在UiPath StudioTM中配置OLDB时，“在屏幕上指示”可查看OLDB，并且拾取UI描述符（如果可用）。如果UI描述符不可用，则UI描述符可由用户通过指示缺少的UI元素来定义。然后可生成UI描述符并在UI对象库中发布。

[0117] 在一些实施例中，UI描述符和UI对象储存库的使用创建了可进一步扩展的关系和数据。在一些实施例中，来自自动化流程的UI步骤可被映射到单个屏幕。例如，包括用户添加和检索一些数据然后点击提交的表单的屏幕可利用包括屏幕捕获和在该屏幕上执行的

每个UI步骤的单个屏幕截图来记录,诸如:(1) 复制数量;(2) 复制数目;(3) 添加名称;以及(4) 点击“确定”

[0118] 在受系统和应用升级影响的升级流程的情况下,可实现“查找引用”服务。查找引用服务可提供从储存库中扫描项目集合并检测正在使用属于某个应用的版本的UI描述符的流程的能力。在一些实施例中,可基于UI描述符的对应屏幕来确定应用版本。此外,可实现用于检测具有来自应用的两个不同版本的匹配元素的UI描述符之间的差异的服务。检测到的流程和UI描述符的差异提供了关于在应用或系统升级时可能中断的流程的信息。这些流程可被自动更新以使用来自新应用版本的UI描述符。

[0119] 在一些实施例中,可使用OLDB分析来增强OLDB UI对象储存库。可实现提取UI描述符的机制。提取的UI描述符数据可在创建/定义时间之前或之后经由OLDB分析发送。此外,在一些实施例中,提取的UI描述符数据可给出新数据集合,该新数据集合可用于使用人工智能(AI)和机器学习(ML)技术来提高来自任何应用的任何版本的任何屏幕的每个UI元素的UI描述符的强度。

[0120] 某些实施例可用于机器人流程自动化(RPA)。图1是图示根据本发明的实施例的RPA系统100的架构图。RPA系统100包括允许开发者设计和实现工作流的设计器110。设计器110可提供用于应用集成以及自动化第三方应用、管理信息技术(IT)任务和商业IT流程的解决方案。设计器110可促进自动化项目的开发,该自动化项目是业务流程的图形表示。简而言之,设计器110促进工作流和机器人的开发和部署。

[0121] 自动化项目通过向开发者提供对在工作流中开发的定制的步骤集合之间的执行顺序和关系的控制来实现基于规则的流程的自动化,在本文中定义为“活动”。设计器110的实施例的一个商业示例是UiPath StudioTM。每个活动可包括一个动作,诸如点击按钮、读取文件、写入日志面板等。在一些实施例中,工作流可以是嵌套式的或嵌入式的。

[0122] 一些类型的工作流可包括但不限于序列、流程图、FSM和/或全局异常处理器。序列可能特别适合于线性流程,支持从一个活动到另一个活动的流动,而不会使工作流变得混乱。流程图可能特别适合更复杂的业务逻辑,通过多个分支逻辑操作符以更多样化的方式实现决策的集成和活动的连接。FSM可能特别适合大型工作流。FSM可在它们的执行中使用有限数目的状态,这些状态由条件(即,转换)或活动触发。全局异常处理程序可能特别适合于在遇到执行错误时确定工作流行为以及调试流程。

[0123] 一旦在设计器110中开发了工作流,业务流程的执行就由指挥器120来协调,指挥器120协调执行设计器110中开发的工作流的一个或多个机器人130。指挥器120的实施例的一个商业示例是UiPath OrchestratorTM。指挥器120促进管理环境中资源的创建、监控和部署。指挥器120可充当第三方解决方案和应用的集成点或聚合点之一。

[0124] 指挥器120可管理机器人130的队列,连接和执行来自中心化点的机器人130。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,但用于开发和测试目的)。有人值守机器人132可由用户事件触发或被调度为自动发生,并且在同一计算系统上与人类一起操作。有人值守机器人132可与指挥器120一起用于中心化流程部署和记录介质。有人值守机器人132可帮助人类用户完成各种任务,并且可由用户事件触发。在一些实施例中,流程不能从这种类型的机器人上的指挥器120开始,

和/或它们不能在锁定的屏幕下运行。在某些实施例中，有人值守机器人132只能从机器人托盘或从命令提示符启动。在一些实施例中，有人值守机器人132应在人类的监管下运行。

[0125] 无人值守机器人134在虚拟环境或物理机器上无人值守地运行，并且可使多个流程自动化。无人值守机器人134可负责远程执行、监控、调度和为工作队列提供支持。在一些实施例中，可从设计器110运行针对所有机器人类型的调试。有人值守和无人值守机器人都可自动化各种系统和应用，包括但不限于大型机、web应用、虚拟机、企业应用（例如，由 **SAP®**、**SalesForce®**、**Oracle®** 等生产的应用），以及计算系统应用（例如，台式和膝上型应用、移动设备应用、可穿戴计算机应用等）。

[0126] 指挥器120可具有各种能力，包括但不限于供应、部署、版本化、配置、排队、监控、记录和/或提供互连性。供应可包括创建和维护机器人130和指挥器120（例如，web应用）之间的连接。部署可包括确保将包版本正确地递送给指定的机器人130以供执行。在一些实施例中，版本化可包括对一些流程或配置的唯一实例的管理。配置可包括机器人环境和流程配置的维护和递送。排队可包括提供队列和队列项的管理。监控可包括跟踪机器人标识数据和维护用户许可。日志记录可包括将日志存储和索引到数据库（例如，SQL数据库）和/或另一种存储机制（例如，提供存储和快速查询大型数据集的能力的 **ElasticSearch®**）。指挥器120可通过充当第三方解决方案和/或应用的中心化通信点来提供互连性。

[0127] 机器人130是运行设计器110中内置工作流的执行代理。（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默认安装微软 **Windows®** 服务控制管理器（SCM）管理的服务。结果，这类机器人130可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。

[0128] 在一些实施例中，机器人130可以以用户模式安装。对于这类机器人130，这意味着在已安装给定机器人130的情况下，它们具有与用户相同的权限。针对高密度（HD）机器人该特征也是可用的，这确保每个机器的最大潜能得到充分利用。在一些实施例中，任何类型的机器人130可被配置在HD环境中。

[0129] 一些实施例中的机器人130分为若干个组件，每个组件专用于特定的自动化任务。一些实施例中的机器人组件包括但不限于SCM管理的机器人服务、用户模式机器人服务、执行器、代理和命令行。SCM管理的机器人服务管理和监控 **Windows®** 会话，并且充当指挥器120和执行主机（即，机器人130在其上执行的计算系统）之间的代理。这些服务被信任并管理机器人130的凭证。本地系统下的SCM启动控制台应用。

[0130] 在一些实施例中，用户模式机器人服务管理和监控 **Windows®** 会话，并且充当指挥器120和执行主机之间的代理。用户模式机器人服务可被信任并管理机器人130的凭证。如果未安装SCM管理的机器人服务，则 **Windows®** 应用可能会自动启动。

[0131] 执行器可在 **Windows®** 会话下运行给定作业（即，它们可执行工作流。执行器可能知道每个监控器的每英寸点数（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。代理可请求启动或停止作业以及更改设置。命令行是服务的客户端。命令行是控制台应用，它可请

求启动作业并等待其输出。

[0132] 如上所述将机器人130的组件拆分,有助于开发者、支持用户和计算系统更轻松地运行、标识和跟踪每个组件正在执行的内容。可通过这种方式为每个组件配置特殊行为,比如为执行器和服务设置不同的防火墙规则。在一些实施例中,执行器可能总是知道每个监控器的DPI设置。结果,工作流可在任何DPI处执行,而不管在其上创建它们的计算系统的配置如何。在一些实施例中,来自设计器110的项目也可独立于浏览器缩放级别。对于DPI无感知或者故意标记为无感知的应用,在一些实施例中可禁用DPI。

[0133] 图2是图示根据本发明的实施例的部署的RPA系统200的架构图。在一些实施例中,RPA系统200可以是图1的RPA系统100,或者可以是图1的RPA系统100的一部分。应注意,在不偏离本发明的范围的情况下,客户端、服务器端或两者可包括任何期望数目的计算系统。在客户端,机器人应用210包括执行器212、代理214和设计器216。然而,在一些实施例中,设计器216可能不在计算系统210上运行。执行器212正在运行流程。若干个业务项目可能同时运行,如图2所示。在该实施例中,代理214(例如,Windows[®]服务)是所有执行器212的单一联系点。该实施例中的所有消息均被记录到指挥器230中,指挥器230经由数据库服务器240、索引器服务器250或两者进一步处理它们。如上关于图1所论述,执行器212可以是机器人组件。

[0134] 在一些实施例中,机器人表示机器名称和用户名之间的关联。机器人可同时管理多个执行器。在支持同时运行的多个交互式会话的计算系统(例如,Windows[®] Server 2012)上,多个机器人可同时运行,每个机器人使用唯一的用户名在单独的Windows[®]会话中运行。这就是上面提到的HD机器人。

[0135] 代理214还负责发送机器人的状态(例如,周期性地发送指示机器人仍在运行的“心跳”消息),并且下载待执行的包的所需版本。在一些实施例中,代理214和指挥器230之间的通信总是由代理214发起。在通知场景中,代理214可打开稍后由指挥器230用来向机器人发送命令(例如,开始、停止等)的WebSocket通道。

[0136] 在服务器端,包括表示层(web应用232、开放数据协议(OData)表示状态传输(REST)应用编程界面(API)端点234以及通知和监控236)、服务层(API实施/业务逻辑238)和持久层(数据库服务器240和索引器服务器250)。指挥器230包括web应用232、OData REST API端点234、通知和监控236以及API实施方式/业务逻辑238。在一些实施例中,用户在指挥器230的界面中(例如,经由浏览器220)执行的大多数动作是通过调用各种API来执行的。在不偏离本发明的范围的情况下,这类动作可包括但不限于启动机器人上的作业、添加/移除队列中的数据、调度作业以无人值守地运行等。Web应用232是服务器平台的可视层。在该实施例中,web应用232使用超文本标记语言(HTML)和JavaScript(JS)。然而,在不偏离本发明的范围的情况下,可使用任何期望的标记语言、脚本语言或任何其他格式。在该实施例中,用户经由浏览器220与来自web应用232的网页交互,以便执行各种动作来控制指挥器230。例如,用户可创建机器人组、将包分配给机器人、分析每个机器人和/或每个流程的日志、启动和停止机器人等。

[0137] 除了web应用232,指挥器230还包括暴露OData REST API端点234的服务层。然而,在不偏离本发明的范围的情况下,可包括其他端点。REST API由web应用232和代理214两者

使用。在本实施例中，代理214是客户端计算机上的一个或多个机器人的监管者。

[0138] 该实施例中的REST API涵盖配置、记录、监控和排队功能性。在一些实施例中，配置端点可用于定义和配置应用用户、许可、机器人、资产、版本和环境。记录REST端点可用于记录不同的信息，诸如错误、机器人发送的显式消息以及其他环境特定信息。如果在指挥器230中使用启动作业命令，则机器人可使用部署REST端点来查询应执行的包版本。排队REST端点可负责队列和队列项管理，诸如向队列添加数据、从队列获取事务、设置事务的状态等。

[0139] 监控REST端点可监控web应用232和代理214。通知和监控API 236可以是REST端点，其用于注册代理214、将配置设置递送到代理214、以及从服务器和代理214发送/接收通知。在一些实施例中，通知和监控API 236也可使用WebSocket通信。

[0140] 在该实施例中，持久层包括一对服务器-数据库服务器240(例如，SQL服务器)和索引器服务器250。该实施例中的数据库服务器240存储机器人、机器人组、相关联流程、用户、角色、时间表等的配置。在一些实施例中，该信息通过web应用232来管理。数据库服务器240可管理队列和队列项。在一些实施例中，数据库服务器240可存储由机器人记录的消息(除了索引器服务器250之外或代替索引器服务器240)。

[0141] 在一些实施例中为任选的索引器服务器250存储并索引机器人记录的信息。在某些实施例中，可通过配置设置来禁用索引器服务器250。在一些实施例中，索引器服务器250使用**ElasticSearch®**，其是开源项目全文搜索引擎。由机器人记录的消息(例如，使用像日志消息或写行之类的活动)可通过(多个)记录REST端点发送到索引器服务器250，在索引器服务器250处它们被索引以供将来使用。

[0142] 图3是图示根据本发明的实施例的设计器310、活动320、330和驱动程序340之间的关系300的架构图。如上所述，开发者使用设计器310来开发由机器人执行的工作流。工作流可包括用户定义的活动320和UI自动化活动330。一些实施例能够标识图像中的非文本视觉成分，这在本文中被称为计算机视觉(CV)。与这些组件相关的一些CV活动可包括但不限于点击、键入、获取文本、悬停、元素存在、刷新范围、突出显示等。在一些实施例中，点击使用例如CV、光学字符识别(OCR)、模糊文本匹配和多锚点来标识元素，并且点击它。键入可使用元素中的上述和类型来标识元素。“获取文本”可标识特定文本的位置，并且使用OCR来扫描它。悬停可标识元素并悬停在其上。元素存在可使用上述技术来检查元素是否存在于屏幕上。在一些实施例中，可能存在可在设计器310中实现的数百甚至数千个活动。然而，在不偏离本发明的范围的情况下，任何数目和/或类型的活动均是可用的。

[0143] UI自动化活动330是以低级代码编写的特殊低级活动(例如，UI活动)的子集，并且促进通过UI层与应用进行交互。在某些实施例中，例如，UI自动化活动300可模拟“通过窗口消息等的用户输入”。UI自动化活动330经由允许机器人与期望软件交互的驱动程序340来促进这些交互。例如，驱动程序340可包括OS驱动程序342、浏览器驱动程序344、VM驱动程序346、企业应用驱动程序348等。

[0144] 驱动程序340可在低级与OS交互，以查找挂钩、监控密钥等。它们可促进与**Chrome®**、**IE®**、**Citrix®**、**SAP®**等的集成。例如，“点击”活动经由驱动程序340在这些不同的应用中执行相同的角色。

[0145] 图4是图示根据本发明的实施例的RPA系统400的架构图。在一些实施例中，RPA系统400可以是或包括图1和/或2的RPA系统100和/或200。RPA系统400包括运行机器人的多个客户端计算系统410。计算系统410能够经由运行在其上的web应用与指挥器计算系统420通信。指挥器计算系统420进而能够与数据库服务器430和任选的索引器服务器440通信。

[0146] 关于图1和图3，应注意，尽管在这些实施例中使用了web应用，但在不偏离本发明的范围的情况下，可使用任何合适的客户端和/或服务器软件。例如，指挥器可运行与客户端计算系统上的非基于web的客户端软件应用通信的服务器端应用。

[0147] 图5是图示根据本发明的实施例的被配置为提供用于RPA的UI描述符、UI对象库、UI对象储存库和/或对象浏览器的计算系统500的架构图。在一些实施例中，计算系统500可以是本文所描绘和/或描述的一个或多个计算系统。计算系统500包括总线505或用于传送信息的其他通信机制，以及耦合到总线505用于处理信息的(多个)处理器510。(多个)处理器510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/或其任何组合。(多个)处理器510还可具有多个处理核心，并且核心中的至少一些可被配置为执行特定功能。在一些实施例中可使用多并行处理。在某些实施例中，(多个)处理器510中的至少一个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，神经形态电路可能不需要冯·诺伊曼计算架构的传统组件。

[0148] 计算系统500进一步包括用于存储信息和将由(多个)处理器510执行的指令的存储器515。存储器515可由随机存取存储器(RAM)、只读存储器(ROM)、闪存、高速缓存、诸如磁盘或光盘的静态存储器或者任何其他类型的非暂态计算机可读介质或其组合的任意组合组成。非暂态计算机可读介质可以是可由(多个)处理器510访问的任何可用介质，并且可包括易失性介质、非易失性介质或两者。该介质也可以是可移动的、不可移动的或两者兼有。

[0149] 此外，计算系统500包括通信设备520，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，在不偏离本发明的范围的情况下，通信设备520可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)、高速下行链路分组接入(HSDPA)、高速上行链路分组接入(HSUPA)、高速分组接入(HSPA)、长期演进(LTE-A)、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、射频标识(RFID)、红外数据协会(IrDA)、近场通信(NFC)、第五代(5G)、新无线电(NR)、其任何组合、和/或任何其他当前现有的或未来实施的通信标准和/或协议。在一些实施例中，在不偏离本发明的范围的情况下，通信设备520可包括一个或多个天线，这些天线是单一号线、阵列天线、相控天线、切换天线、波束成形天线、波束操纵天线、其组合，和/或任何其他天线配置。

[0150] (多个)处理器510还经由总线505耦合到显示器525，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、面内切换(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525可被配置为使用电阻、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、多点触摸显示器等。在不偏离本发明的范围的情况下,可使用任何合适的显示设备和触觉I/O。

[0151] 键盘530和诸如计算机鼠标、触摸板等的光标控制设备535进一步耦合到总线505,以使用户能够与计算系统500交互。然而,在某些实施例中,可不存在物理键盘和鼠标,并且用户可仅通过显示器525和/或触摸板(未示出)与设备交互。根据设计选择,可使用任何类型和组合的输入设备。在某些实施例中,不存在物理输入设备和/或显示器。例如,用户可经由与其通信的另一个计算系统远程地与计算系统500交互,或者计算系统500可自主操作。

[0152] 存储器515存储由(多个)处理器510执行时提供功能性的软件模块。这些模块包括计算系统500的操作系统540。这些模块进一步包括UI对象操作模块545,其被配置为执行本文所描述的所有或部分流程或其衍生物。计算系统500可包括一个或多个包括附加功能性的附加功能性模块550。

[0153] 本领域技术人员将理解,在不偏离本发明范围的情况下,“系统”可体现为服务器、嵌入式计算系统、个人计算机、控制台、个人数字助理(PDA)、手机、平板计算设备、量子计算系统或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中的一个示例。实际上,本文所公开的方法、系统和设备可按符合计算技术(包括云计算系统)的本地化和分布式形式来实施。

[0154] 应注意,本说明书中描述的一些系统特征已作为模块呈现,以便更具体地强调其实施方式独立性。例如,模块可被实施为硬件电路,该硬件电路包括定制的超大规模集成电路(VLSI)电路或门阵列、诸如逻辑芯片、晶体管或其他分立组件的现成半导体。模块也可在可编程硬件设备,诸如现场可编程门阵列、可编程阵列逻辑、可编程逻辑设备、图形处理单元等中实施。

[0155] 模块也可至少部分在软件中实施,以供各种类型的处理器执行。可执行代码的标识单元可例如包括计算机指令的一个或多个物理或逻辑块,该计算机指令可例如被组织为对象、流程或功能。然而,所标识的模块的可执行代码不需要在物理上位于一起,而是可包括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,构成该模块并实施该模块的所述目的。此外,在不偏离本发明的范围的情况下,模块可存储在计算机可读介质上,该计算机可读介质可以是例如硬盘驱动程序、闪存设备、RAM、磁带和/或用于存储数据的任何其他这类非暂态计算机可读介质。

[0156] 实际上,可执行代码模块可以是单个指令或多个指令,并且甚至可分布在若干个不同代码段、不同程序和若干个存储设备中。类似地,操作数据在本文可在模块内被标识和图示,并且可按任何合适的形式被体现并被组织在任何合适类型的数据结构内。操作数据可作为单个数据集来收集,或者可分布在包括不同存储设备的不同位置上,并且可至少部分地仅仅作为系统或网络上的电子信号而存在。

[0157] 一些实施例将对应于来自某个版本的应用的屏幕的UI描述符集合的对象库存储在数据库中。当用户开发RPA UI自动化时,对象库可由RPA设计器应用(例如,UiPath StudioTM)创建。这可有助于创建可用于适应后设计时更改(诸如应用的新版本)的更具可扩展性和适应性的UI对象集合。

[0158] 考虑员工使用一个或多个RPA机器人在其计算机上运行的软件应用上执行任务的

情况。然后,信息技术(IT)部门升级RPA机器人与其交互的软件应用。如果公司的RPA团队没有提前对称为RPA机器人工作流的经修改的UI描述符进行更改,则一些RPA流程现在可能会中断,因为RPA机器人由于新版本中的更改(例如,更改的UI元素功能性和/或外观、UI元素被完全移除或移动到新屏幕等)而无法再找到应用中的某些UI元素。如果两个版本的UI元素的UI描述符均已存在,则可在RPA工作流的适当活动中将新版本的UI描述符与先前版本的UI描述符互换,然后RPA机器人应在新版本上工作。

[0159] 图6A是图示根据本发明的实施例的RPA设计器应用600的屏幕截图。RPA设计器应用600包括主RPA开发窗格610,该主RPA开发窗格包括RPA工作流612(在该示例中,序列)。“摘录”窗格620包括可重用并适用于多个工作流的可重用组件或公共自动化项目。属性窗格640显示RPA工作流612的所选择的活动614的属性。

[0160] 在该实施例中,RPA工作流612在Windows®7计算器的标准屏幕(或窗口)670上执行一系列动作。参见图6B。用户可使用视图选项卡672从用于不同计算器设置的不同屏幕中进行选择。RPA工作流612中的动作序列是点击按钮2、点击加号按钮、点击按钮3、点击等于按钮、获取结果的文本、点击清除按钮、以及经由“写入行”操作来写入结果。图6C示出了提纲窗格630中序列的放大视图。

[0161] 图6D示出了摘录窗格620的放大视图。摘录窗格620包括工作流摘录,其可以是开发者可在RPA设计器应用600中调用/包括在他或她自己的工作流(例如,RPA工作流612)中的现成工作流。在一些实施例中,摘录窗格620可提供沙箱,在该沙箱中,团队公司、所有感兴趣的开发者或任何其他开发者群体可保存要自动化的元素的UI描述符。摘录窗格620包括按应用排列的UI描述符621。在计算器应用622下,Windows®10选择符被组织在Windows®10版本623、标准窗口624、目标UI元素625和相应的UI元素选择符626之下。

Windows®7选择符被组织在Windows®7版本623A、标准窗口624A、目标UI元素625A以及相应的UI元素选择符626A之下。尽管在该实施例中使用了选择符,但在一些实施例中,除了UI元素选择符之外或者代替UI元素选择符,可包括CV描述符、统一目标描述符或者两者。

[0162] 图6E和图6F示出了RPA工作流序列612的活动。Windows®7标准计算器屏幕上与活动中的每一个相对应的部分的屏幕截图会针对该活动进行显示并突出显示(如果适用)。例如,可能已使用UiPath Studio™中的“指示”功能性选择了活动的相应UI元素,这将在下文中关于对活动使用多个UI描述符类型进行更详细的论述。通过这种方式,用户可直观地验证RPA工作流是否应与期望的UI元素进行交互。通过点击与UI元素交互的活动的按钮615,出现菜单616。参见图6G。用户可通过选择添加到UI对象选项618来将相应的UI对象添加到对象浏览器。

[0163] 图6H图示了具有在属性窗格640中示出的点击活动的属性的RPA设计器应用600。图6I示出了属性窗格640的放大视图。在属性窗格640的底部,用户可经由UI对象浏览器选项卡652选择UI对象浏览器。

[0164] 图6J图示了具有UI对象浏览器窗格650的UI描述符视图的RPA设计器应用600,并且图6K示出了具有UI描述符视图的UI对象浏览器窗格650的放大视图。UI描述符视图包括项目UI描述符654A和UI对象库656A。在图6K中,还没有为项目选择UI描述符。

[0165] 为了添加UI描述符,开发者可使用摘录窗格620添加这些描述符。例如,开发者可在摘录窗格620中的**Windows® 7**版本623A上右击,这使菜单627出现。开发者可选择将这些UI描述符添加到项目中,这使它们出现在UI对象浏览器窗格650中的项目UI描述符654A之下,这将它们显示为UI描述符658A。参见图6L。在图6M所示的UI对象浏览器窗格650的UI活动视图中,示出了项目654B和UI活动656B。尽管RPA工作流可具有不同类型的各种活动,但UI活动视图列出了UI活动656B,以便开发者可容易地查看它们。UI活动656B可被提取并具有与摘录窗格620中所示的类似的结构。在某些实施例中,已开发并在库中共享的所有UI描述符可显示在摘录窗格620中,而只有由UI活动使用的UI描述符出现在UI对象浏览器窗格650的UI活动视图中。

[0166] 一旦选择符是在对象浏览器窗格650中,用户可将给定选择符拖放至期望的活动中,如图6N所示。这提供了项目级的可重用性。在一些实施例中,除了选择符之外或者代替选择符,可将诸如CV描述符、统一目标描述符等其他UI描述符拖放到期望的活动中。一旦选择符被放入相应的活动中,RPA工作流612就准备好由RPA机器人执行以与相应的UI元素交互。

[0167] 考虑开发者希望对**Windows® 10**计算器(见图6O的**Windows® 10**计算器680)执行相同的RPA流程的情况。对于与各种UI元素交互的活动,运行为**Windows® 7**计算器设计的RPA流程可能会失败,因为**Windows® 10**计算器的UI元素与**Windows® 7**计算器的UI元素有很大不同。参见图6B。如果已为**Windows® 10**计算器开发了选择符,并且在摘录窗格620中的UI描述符的已发布库中可用,如本示例中的情况,则用户可删除用于**Windows® 7**计算器的选择符,并且添加用于**Windows® 10**计算器的选择符,然后这些选择符出现在UI对象浏览器窗格650的UI活动视图中。根据本发明的实施例,参见具有**Windows® 10**计算器选择符的图。

[0168] 当目标应用设置为**Windows® 10**计算器时,每个活动的屏幕截图部分和相应的选择符在RPA工作流612中更新。参见图6Q。RPA工作流612将无法在**Windows® 10**计算器上正确执行。这提供了项目级的可重用性。然而,选择符在这个项目中是特定的,尚未作为UI对象库在全球范围内使用。

[0169] 为了“升级”计算器应用,将UI元素的UI描述符更改为新版本的应用的UI描述符。例如,包括这些UI描述符的NuGet™包可被创建、上传,然后被检索。因此,当UI元素的依赖项改变时,UI描述符将被更新。

[0170] 为了提供全局级的可重用性,使所有流程均可使用UI描述符,可使用UI对象库656A。在该实施例中,通过点击发布按钮658(见图6P),出现发布UI对象库窗口660。参见图6R。这允许来自项目的选择符和/或其他UI描述符被发布到包中(例如,发布到NuGet™包中),该包可在经由管理包窗口662发布之后被其他开发者访问和用于其他RPA流程。参见图6S。安装该计算器描述符包作为项目的依赖项,使选择符/UI描述符出现在UI对象浏览器窗格650中的库656A之下。参见图6T。然后,UI对象库可用于给定的项目。作为最佳实践,用于与UI元素交互的选择符和/或其他UI描述符仅在需要重用时才可作为依赖项从UI对象库中

导入。然而,如果给定的选择符或其他UI描述符是特定于给定项目的,并且不太可能被重用,则某些选择符/UI描述符可以是该项目本地创建,而其他选择符/UI描述符可从UI对象库导入。

[0171] 图7A是图示根据本发明的实施例的RPA设计器应用700的屏幕截图。RPA设计器应用700包括主RPA开发窗格702,该主RPA开发窗格包括RPA工作流710(在该示例中,包括不同UI元素标识活动的序列)。项目窗格704包括当前项目的依赖项。UI描述符窗格706当前不包括对项目UI描述符和UI对象库的选择。

[0172] 图7B是图示RPA工作流710的放大视图。活动720将使用基于选择符的标识,活动730将使用基于CV的标识,并且活动740将使用基于统一目标的标识。下面描述了该示例的活动720、730、740的配置。

[0173] 图7C图示了可由UiPath StudioTM修改的Windows[®] 10计算器750。当用户为基于选择符的标识活动720选择“在屏幕上指示”721时(参见图7B),开发者将鼠标移到其上的UI的识别元素被突出显示。例如,在图7C中,9按钮752被突出显示。当用户点击按钮752时,包括所选择的UI元素的屏幕截图722出现在基于选择符的标识活动720中。参见图7D。

[0174] 通过点击基于选择符的标识活动720的按钮724,出现菜单725。参见图7E。选择“编辑选择符”选项打开选择符编辑器726。参见图7F。选择符编辑器726允许开发者编辑选择符的特性,并且验证选择符在Windows[®] 10计算器750中找到相应的UI元素(即,9按钮752)。选择“突出显示”使相应的UI元素在Windows[®] 10计算器750上突出显示。在创建了选择符之后,开发者可点击基于选择符的标识活动720的按钮724,并且选择菜单725上的“添加到UI对象”选项,这使出现添加到UI对象库窗口728。参见图7G。然后,开发者可将为9按钮创建的选择符添加到项目的UI对象库中。选择符现在出现在UI描述符窗格706中。参见图7H。

[0175] 为了配置基于CV的标识活动730,在该实施例中,用户应首先指示将在其上执行CV的屏幕。当用户选择“在屏幕上指示”731(参见图7B)时,用户点击Windows[®] 10计算器750的UI,这使UiPath StudioTM检测元素(使用CV)和标签(使用所选择的OCR引擎732——在这种情况下是Windows[®] OCR),使消息733出现。参见图7I。一旦检测到Windows[®] 10计算器750的UI中的元素和标签,检测到的UI元素734的屏幕截图出现在基于CV的标识活动730中。参见图7K。可使用CV点击活动735,其可被配置为使用CV来标识9按钮。可使用搜索界面708将该活动添加为基于CV的标识活动730内的嵌套活动,该搜索界面可在开发者点击给定活动中的加号图标时出现。参见图7J。

[0176] 当开发者点击针对CV点击活动735的范围736上的指示时,Windows[®] 10计算器750与CV点击界面754一起出现。参见图7L。然而,当用户点击9按钮752时,该按钮不是使用CV唯一地标识的。参见图7M。CV点击界面754向开发者显示发现重复匹配的消息,并且要求开发者添加锚点。开发者然后添加9按钮752的锚点(例如,使用锚点选择功能性的所选锚点756、757和候选锚点758),直到9按钮752基于其与Windows[®] 10计算器750中的其他锚点UI元素的关系被唯一地标识为目标UI元素。在9按钮752的CV描述符使用目标和(多个)锚点唯一地标识它之后,屏幕截图737出现在CV点击活动735中。参见图7O。在图7O中,屏幕截图737示出了没有正确选择目标和锚点的版本,但这仅作为示例示出。在某些实施例中,CV描

述符可被保存,而无需将其配置为唯一地标识目标UI元素。

[0177] 开发者可使用属性选项卡760查看具有描述符属性的CV点击。见图7P。这提供了CV描述符的各种变量。然而,与选择符不同,用于CV描述符的选择符的底层特性可能是复杂的,并且与AI/ML模型到图像的应用有关。因此,尽管提供了CV选择符,但在一些实施例中,它可能不容易被开发者访问或修改。相反,开发者可重新指示目标UI元素,以便编辑CV描述符/选择符。这也意指在一些实施例中,甚至非技术用户也可指示目标/锚点。屏幕上的UI元素和开发者希望点击的UI元素的集合可被保存为基于CV的标识活动730的一部分。

[0178] 关于基于统一目标的标识活动740,当开发者选择“指示应用”741(见图7B)时,用户点击Windows® 10计算器750的UI,这使UiPath Studio™捕获UI的屏幕截图743,并且在基于统一目标的标识活动740中显示屏幕截图743。参见图7Q。还示出了与屏幕截图743相关联的应用的应用路径744,并且如果需要,开发者可添加应用参数745。do活动742嵌套在基于统一目标的标识活动740中。当开发者点击加号图标时,搜索界面708出现。参见图7R。开发者然后可搜索期望的点击功能性。

[0179] 所选择的点击选项使嵌套的点击活动746出现在do活动742中。参见图7S。点击活动746包括在屏幕上指示按钮747,该按钮使得开发者能够指示将在屏幕上选择和点击的目标。开发者还可指定点击类型和将点击的鼠标按钮。在这种情况下,指定点击一次左键。

[0180] 点击在屏幕上指示按钮747,出现统一目标选择选项窗口770。参见图7T。统一目标基于图像添加目标和/或锚点,并且在幕后使用基于选择符的框架。将鼠标悬停在9按钮上使目标UI元素提纲772出现。当用户点击9按钮时,它被选择为目标,并且该状态被指示在目标UI元素提纲772上。参见图7U。各种选项图标也出现,包括锚点指定图标774。在该实施例中,UiPath Studio™试图自动找到合适的锚点,但无法找到。

[0181] 开发者指定锚点776(在这种情况下,8按钮),这些元素的组合允许统一目标逻辑使用目标和锚点8按钮的组合唯一地标识目标9按钮。参见图7V。这使统一目标选择选项窗口770确认目标和锚点是可以的,并且设计器可验证目标和锚点选择符、确认选择或者取消。在此,开发者选择验证目标和锚点选择符,并且统一目标选择选项窗口770确认验证成功,验证准确度得分为100%,如验证得分通知778中所示。参见图7W。在指示目标和锚点选择符正常之后,出现目标和锚点的屏幕截图748。参见图7X。在某些实施例中,CV技术可在统一目标框架内使用。

[0182] 在一些实施例中,可采用图像匹配来查找UI元素。例如,在图7Y中已添加了点击图像活动780。当开发者选择“在屏幕上指示图像”782时,开发者可定义用于图像搜索的图像。在图7Z中,开发者在9按钮周围拖动框784,并且9按钮的屏幕截图786出现在点击图像活动780中。点击图像活动的属性出现在属性选项卡760中。参见图7a。也生成图像的选择符,如图7b的选择符编辑器726所示。选择符捕获开发者从中获取快照的应用。

[0183] 在某些实施例中,可使用选择符和OCR的组合。图7c示出点击OCR文本活动790,该点击OCR文本活动包括开发者可手动输入文本以进行搜索的字段794和OCR引擎796。在该实施例中,当用户点击“在屏幕上指示”792时,用户以与基于选择符的方法类似的方式选择UI元素(参见图7C)。在选择UI元素之后,所选择的UI元素的屏幕截图798出现在点击OCR文本活动790中。参见图7d。然而,与基于选择符的方法不同,选择符和OCR方法还使用OCR来标识UI元素。例如,尽管许多计算器按钮具有相同的形状,但只有一个按钮具有每个数字。

[0184] 图8是图示根据本发明的实施例的用于生成基于选择符的活动的流程800的流程图。该流程开始于在810生成基于选择符的活动。该活动可点击按钮、输入文本、打开菜单和选择选项等。在820接收在屏幕上指示命令，并且在830检测应用。在一些实施例中，这可包括自动检测应用、检测应用的版本和/或检测应用内的屏幕。如果应用没有被自动检测，检测应用可包括请求用户提供应用名称和版本。例如，如果检测到多个应用和/或用户可手动输入该信息，则可向用户提供下拉列表以提供应用名称和应用版本。

[0185] 可提供一系列活动，这些允许开发者创建期望的RPA流程。一些活动可能允许用户使用来指示屏幕/应用。当用户希望执行自动化时，用户可指示活动应与其交互的屏幕、屏幕上的UI元素等。某些活动可能会打开应用。这可能包括打开桌面应用、使用URL导航等。

[0186] 在840检测应用屏幕上的UI元素。在一些实施例中，可以为用户突出显示检测到的元素以生成UI模型。检测到的UI元素可被组织为分类或树，作为屏幕的模型。然而，在某些实施例中，可使用诸如本体的非分类学方法。本体可定义屏幕之间的某些结构和逻辑关系。例如，如果在一个屏幕上点击“提交”按钮，则这可能会使另一个屏幕出现至少一些不同的UI元素。此外，屏幕的某些UI元素和/或部分可以是多个屏幕所共有。例如，在Microsoft Outlook[®]中，左栏导航子屏幕传统地不会改变，并且可在屏幕之间重用。知道哪些子屏幕和/或UI元素没有改变可减少处理时间。

[0187] 在850，从用户接收活动的UI元素的指示。在一些实施例中，这可包括点击期望的UI元素。在860，基于UI元素的最佳匹配，使用驱动程序（例如，驱动程序340）为所指示的UI元素生成开箱即用选择符，并且相应地配置活动。

[0188] 图9是图示根据本发明的实施例的用于生成基于CV描述符的活动的流程900的流程图。该流程开始于在910生成基于CV描述符的活动。在920，接收在屏幕上指示命令和用户对要自动化的屏幕的选择。例如，在一些实施例中，用户可选择整个应用界面。在930检测应用，并且在940使用CV检测应用屏幕上的UI元素。

[0189] 在950，接收为交互指定UI元素的命令，显示具有由CV检测到的可选UI元素的界面。然后在960，从用户接收对目标UI元素的选择。如果在970可唯一地标识目标，则在980生成CV描述符并相应地配置活动。然而，如果在970不能唯一地标识目标，则在990需要锚点指定并从用户接收锚点指定，直到可唯一地标识目标。然后，流程进行到步骤980，以生成CV描述符并配置活动。

[0190] 图10是图示根据本发明的实施例的用于生成基于统一目标的活动的流程1000的流程图。在1020，接收指示应用命令和用户对要自动化的屏幕的选择。例如，在一些实施例中，用户可选择整个应用界面。例如，在1030检测应用，并且在1040使用驱动程序检测应用屏幕上的UI元素。

[0191] 在1050，接收为交互指定UI元素的命令，并且显示具有由驱动程序检测到的可选UI元素的界面。然后在1060，从用户接收对目标UI元素的选择。如果在1070仅使用该元素的选择符可唯一地标识目标，则在1080生成统一目标描述符（例如，选择符）并相应地配置活动。然而，如果在1070不能唯一地标识目标，则需要锚点指定并从用户接收锚点指定，直到在1090可唯一地标识目标。在一些实施例中，这可包括通过确定验证得分来验证目标和（多个）锚点。该流程然后前进到步骤1080，以生成统一目标描述符（例如，用于目标和（多个）锚点以及它们之间的位置/几何关系的选择符）并配置该活动。

[0192] 为了确定UI元素是否被唯一地标识,强度计算(例如,验证得分)可用于基于统一的目标UI描述符确定相信目标UI元素匹配得有多好。如果强度计算低于阈值(例如,低于97%、低于95%等等),则有可能无法在运行时如期望的那样准确地确定UI元素。可接受的准确度可能因RPA工作流所完成的(多个)任务而异。低于阈值的UI元素目标可能需要锚点来进行唯一地标识。这些锚点可被自动确定并显示给用户,或者用户可将鼠标悬停在UI元素上来选择锚点。在一些实施例中,可向用户显示由于添加给定锚点而产生的统一目标描述符强度。如果在添加锚点之后仍未达到阈值,则用户可能需要继续添加附加的锚点,直到目标UI元素达到阈值。

[0193] 锚点是屏幕上的其他UI元素,可用于帮助唯一地标识屏幕上的目标UI元素。例如,如果一个屏幕上包括多个文本字段,则单独搜索一个文本字段不足以唯一地标识给定的文本字段。因此,一些实施例寻找附加信息以便唯一地标识给定的UI元素。使用文本字段示例,用于输入名字的文本字段可出现在标签“名字”的右侧。这个名字标签可被设置为“锚点”,以帮助唯一地标识作为“目标”的选择符的文本字段。

[0194] 在一些实施例中,目标和锚点之间的各种位置和/或几何关联可以潜在地在容差内使用,以唯一地标识目标。例如,锚点和目标的边界框的中心可用于定义线段。然后,可要求该线段具有在容差内的特定长度和/或在容差内的斜率,以使用目标/锚点对唯一地标识目标。

[0195] 在一些情况下,单个锚点可能不足以唯一地标识屏幕上的目标元素。例如,考虑用于输入名字的两个文本字段出现在屏幕上不同位置的相应标签“名字”的右侧的情况。在这类情况下,一个或多个附加锚点对于唯一地标识给定目标可能是有用的。锚点和目标之间的几何属性(例如,线段长度、角度和/或具有容差的相对位置)可用于唯一地标识目标。可要求用户继续添加锚点,直到目标的匹配强度超过阈值。

[0196] 在一些实施例中,可自动分配一个或多个锚点,直到强度计算超过阈值,而不是向用户标记和显示UI元素和锚点候选。例如,如果阈值是97%,并且给定的UI元素具有90%的匹配强度,则系统可继续添加锚点,直到匹配强度达到或超过阈值。在一些实施例中,这可通过以下方式来完成:确定对选择符增加匹配强度最多的锚点,如果仍然低于阈值,则添加该锚点,在添加锚点之后确定其他元素的匹配强度增加,添加使匹配强度最高增加的锚点,并且重复直到满足阈值。

[0197] 图11是图示根据本发明的实施例的用于生成图像匹配活动的流程1100的流程图。该流程开始于在1110生成图像匹配活动。该活动可基于图像匹配技术点击按钮、输入文本、打开菜单和选择选项等。在1120接收在屏幕上指示命令,并且在1130检测应用。

[0198] 在1140,接收对要执行图像匹配的应用区域的选择。在不偏离本发明的范围的情况下,这可包括用户在感兴趣的UI元素周围绘制框、使用套索工具、绘制自由形式的选择或任何其他选择机制。然后,在1150,为所选图像生成图像匹配选择符,并且相应地配置活动。

[0199] 图12是图示根据本发明的实施例的用于生成基于选择符和OCR的活动的流程1200的流程图。该流程开始于在1210生成基于选择符和OCR的活动。该活动可点击按钮、输入文本、打开菜单和选择选项等。在1220接收在屏幕上指示命令,并且在1230检测应用。在1240,还检测应用屏幕上的UI元素。

[0200] 在1250,从用户接收活动的UI元素的指示。在一些实施例中,这可包括点击期望的

UI元素。在1260,使用基于UI元素的最佳匹配的驱动程序为所指示的UI元素生成开箱即用选择符,并且相应地配置活动。在1270,接收对OCR引擎的选择,并且相应地配置活动。当基于选择符和OCR的活动作为RPA流程的一部分执行时,它会检查是否找到了UI元素的选择符,并且验证UI元素的文本是否与使用OCR的活动中指定的文本相匹配,而不是单独搜索选择符。

[0201] 图13是图示根据本发明的实施例的用于生成使用多种标识技术的RPA工作流的流程1300的流程图。该流程开始于在1310生成UI元素标识和交互活动。这些活动可以是一个或多个基于选择符的标识活动、基于CV的标识活动、统一目标标识活动、图像匹配标识活动、基于选择符和OCR的标识活动、任何其他合适的标识活动、其任何组合等。

[0202] 然后,在1320,将UI元素标识和交互活动配置为逻辑关系。逻辑关系可以是顺序的、有条件的(例如,如果第一个活动没有成功地唯一检测到UI元素,则使用第二个,然后第三个,等等,直到唯一地标识了UI元素或者已尝试了所有活动),并行的等等。在1330,生成实施包括UI元素标识和交互活动的工作流的RPA机器人,并且在1340,执行RPA机器人来检测UI元素并与之交互。

[0203] 图14是图示根据本发明的实施例的用于创建UI对象库和配置项目以与新版本的应用一起工作的流程1400的流程图。如果先前已在1410创建了一个或多个UI对象库,则该流程可开始于从UI对象储存库加载一个或多个UI对象库。然后,在1420,从一个或多个UI对象库中创建和/或添加用于与活动的相应UI元素进行交互的UI描述符,以与具有改变的UI的应用的新版本进行交互。

[0204] 一旦更新了适当的UI描述符,则在1430,在UI对象储存库中发布包括UI描述符的UI对象库。UI对象库还可包括应用、屏幕以及来自活动与其交互的应用的至少一些UI元素。然而,在某些实施例中,可包括未在RPA工作流中具体使用的其他UI元素和/或UI描述符。然后,在1440,在UI对象储存库中发布或重新发布UI对象库,以在其他RPA工作流中使用和/或由其他开发者使用。

[0205] 如果UI对象库中缺少UI对象,或者活动的UI描述符不再适用于新版本的应用,则用户可添加这些对象。所创建的UI对象库以及UI对象储存库中可能的其他UI对象库可在设计器应用中向用户显示。在一些实施例中,对象库的发布可作为全局对象库数据库(OLDB)或本地OLDB的一部分来执行。例如,全局OLDB UI对象储存库可以是全局共享的数据库(卓越中心(CoE)级别)。本地OLDB UI对象储存库可以是本地共享的数据库(本地机器级别)。在一些实施例中,UI对象库可被推送并发布到服务,诸如UiPathTM服务或Go!服务。如果需要,这允许与世界共享UI对象库集合。UI对象储存库(例如,OLDB)的存储形式应在架构/技术级别上决定。

[0206] 根据本发明的实施例,图8至14中执行的流程步骤可由计算机程序执行,该计算机程序编码用于(多个)处理器以执行图8-14中描述的(多个)流程的至少一部分的指令。计算机程序可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程序、闪存设备、RAM、磁带和/或用于存储数据的任何其他这类介质或介质的组合。计算机程序可包括用于控制计算系统的(多个)处理器(例如,图5的计算系统500的(多个)处理器510)以实施图8至14中描述的全部或部分流程步骤的编码指令,该编码指令也可存储在计算机可读介质上。

[0207] 计算机程序可按硬件、软件或混合实施方式实施。计算机程序可由彼此可操作通信的模块组成，并且被设计成将信息或指令传递给显示器。计算机程序可被配置为在通用计算机、ASIC或任何其他合适的设备上运行。

[0208] 容易理解的是，本发明各种实施例的组件，如本文附图中一般描述和说明的，可按各种不同配置来布置和设计。因此，如附图中所示，本发明的实施例的详细描述不旨在限制所要求保护的本发明的范围，而仅仅是本发明的所选实施例的表示。

[0209] 在一个或多个实施例中，贯穿本说明书描述的本发明的特征、结构或特性可按任何合适的方式组合。例如，在整个说明书中对“某些实施例”、“一些实施例”或类似语言的引用意指结合该实施例描述的特定特征、结构或特性包括在本发明的至少一个实施例中。因此，在整个说明书中出现的短语“在某些实施例中”、“在一些实施例中”、“在其他实施例中”或类似语言不一定均指同一组实施例，并且在一个或多个实施例中，所描述的特征、结构或特性可按任何合适的方式组合。

[0210] 应注意，在整个本说明书中对特征、优点或类似语言的引用并不意指可用本发明实现的所有特征和优点应在本发明的任何单个实施例中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指结合实施例描述的特定特征、优点或特性被包括在本发明的至少一个实施例中。因此，贯穿本说明书的特征和优点的论述以及类似的语言可以但不一定指同一实施例。

[0211] 此外，在一个或多个实施例中，本发明的所述特征、优点和特性可按任何合适的方式进行组合。相关领域的技术人员将认识到，本发明可在没有特定实施例的一个或多个具体特征或优点的情况下实施。在其他情况下，在某些实施例中可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。

[0212] 本领域普通技术人员将容易理解，如上所论述的本发明可通过不同顺序的步骤和/或不同于所公开配置的硬件元件实施。因此，尽管已基于这些优选实施例描述了本发明，但对于本领域的技术人员来说，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，为了确定本发明的尺度和界限，应参考所附权利要求。

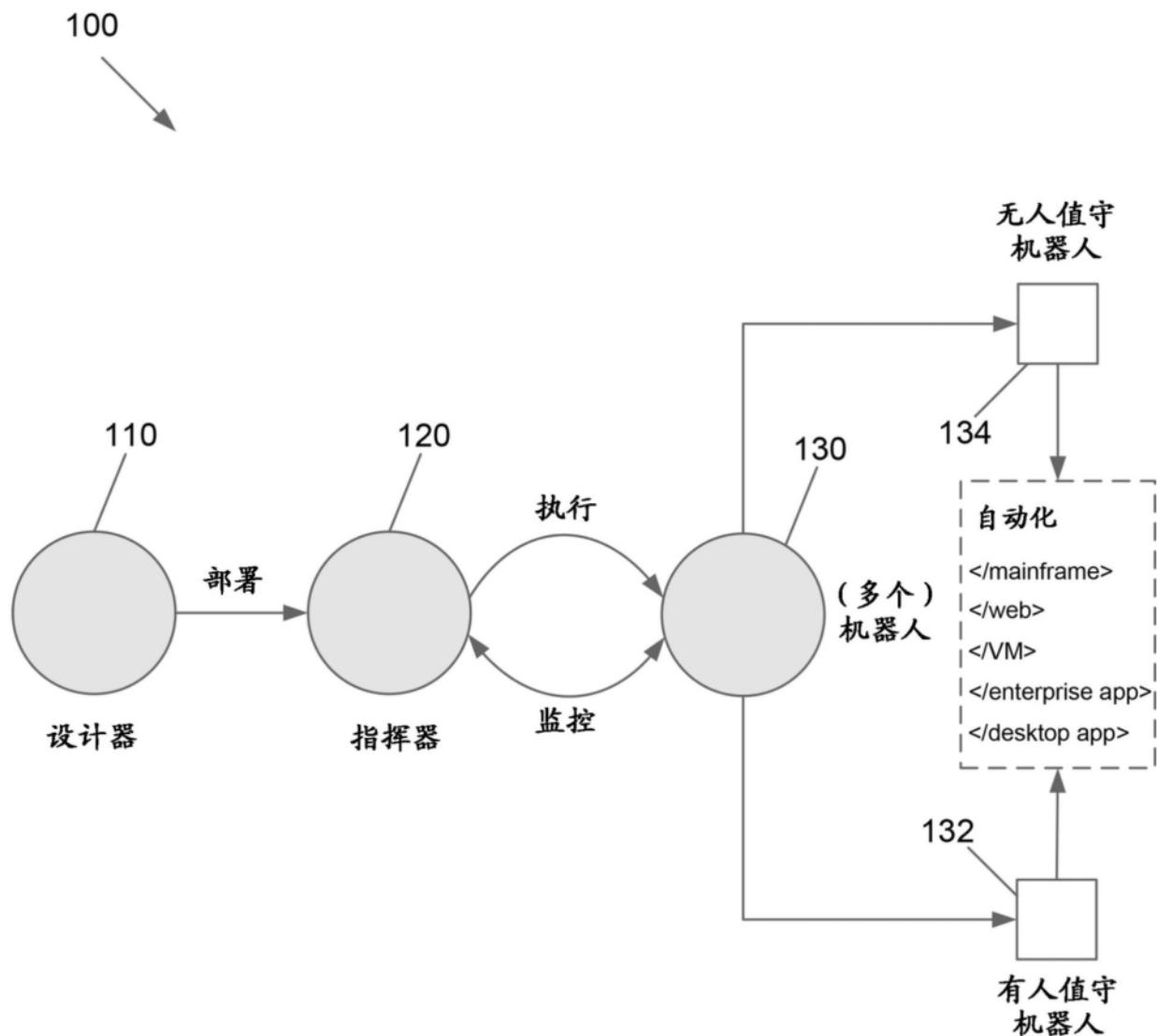


图1

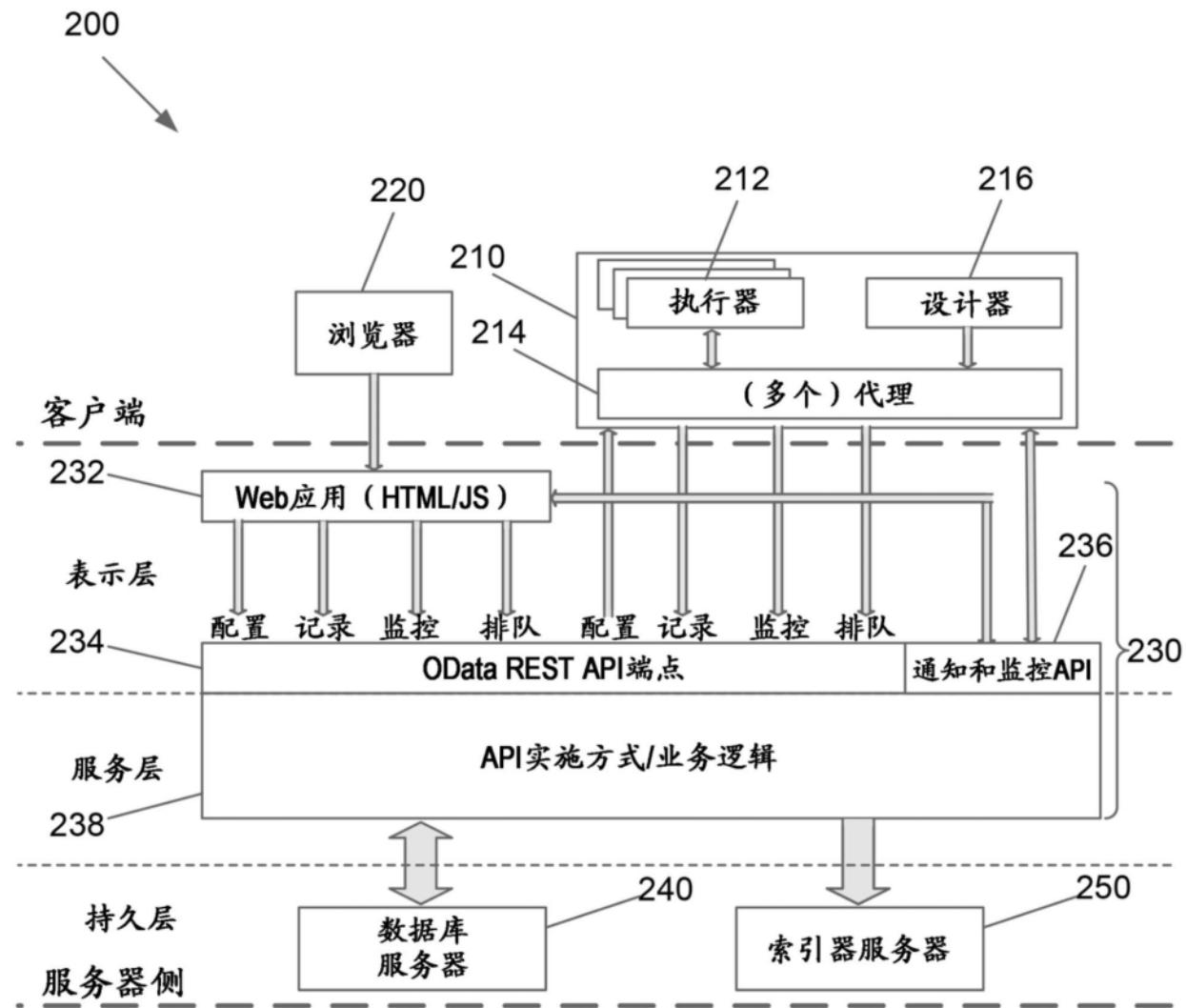


图2

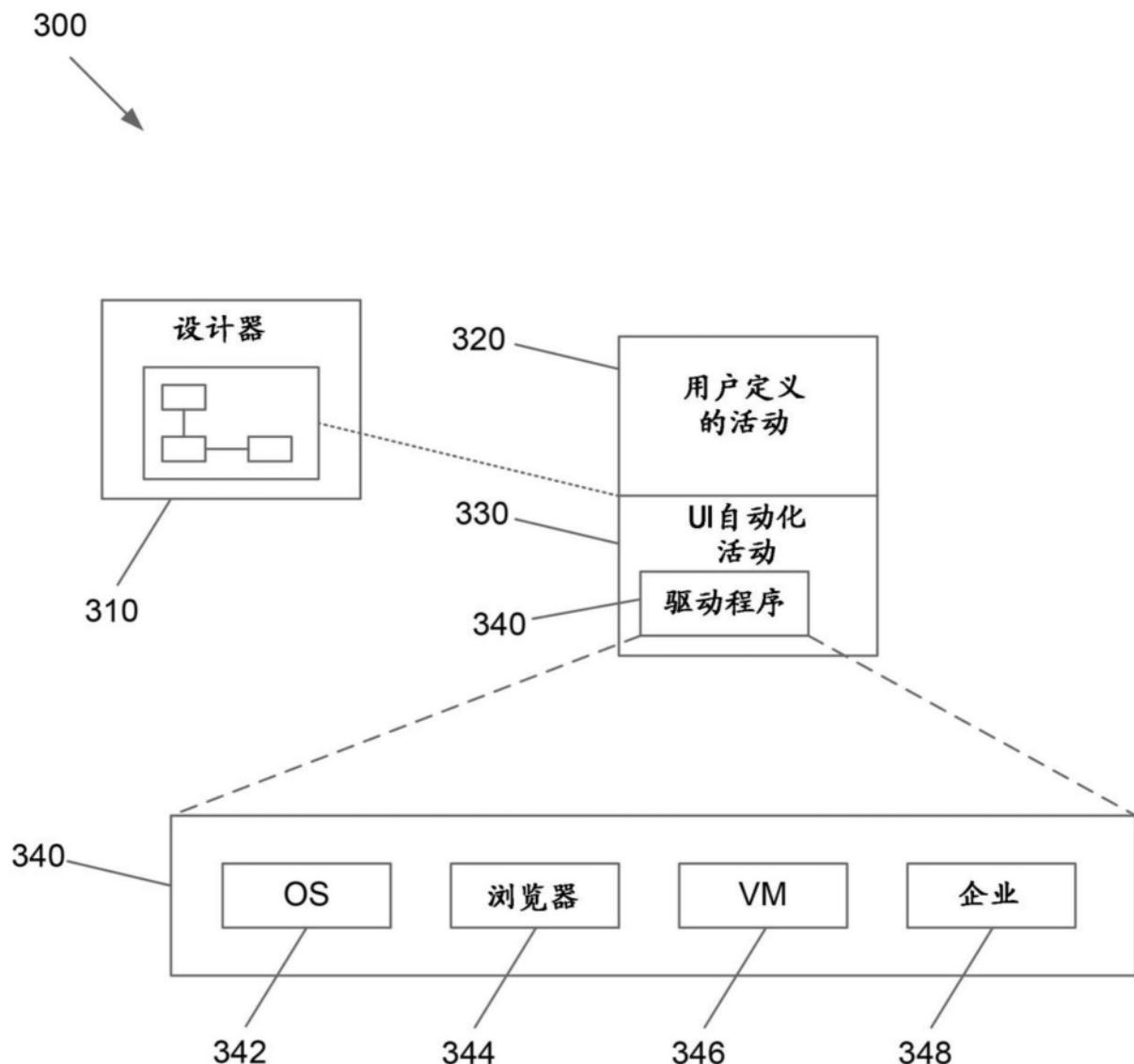


图3

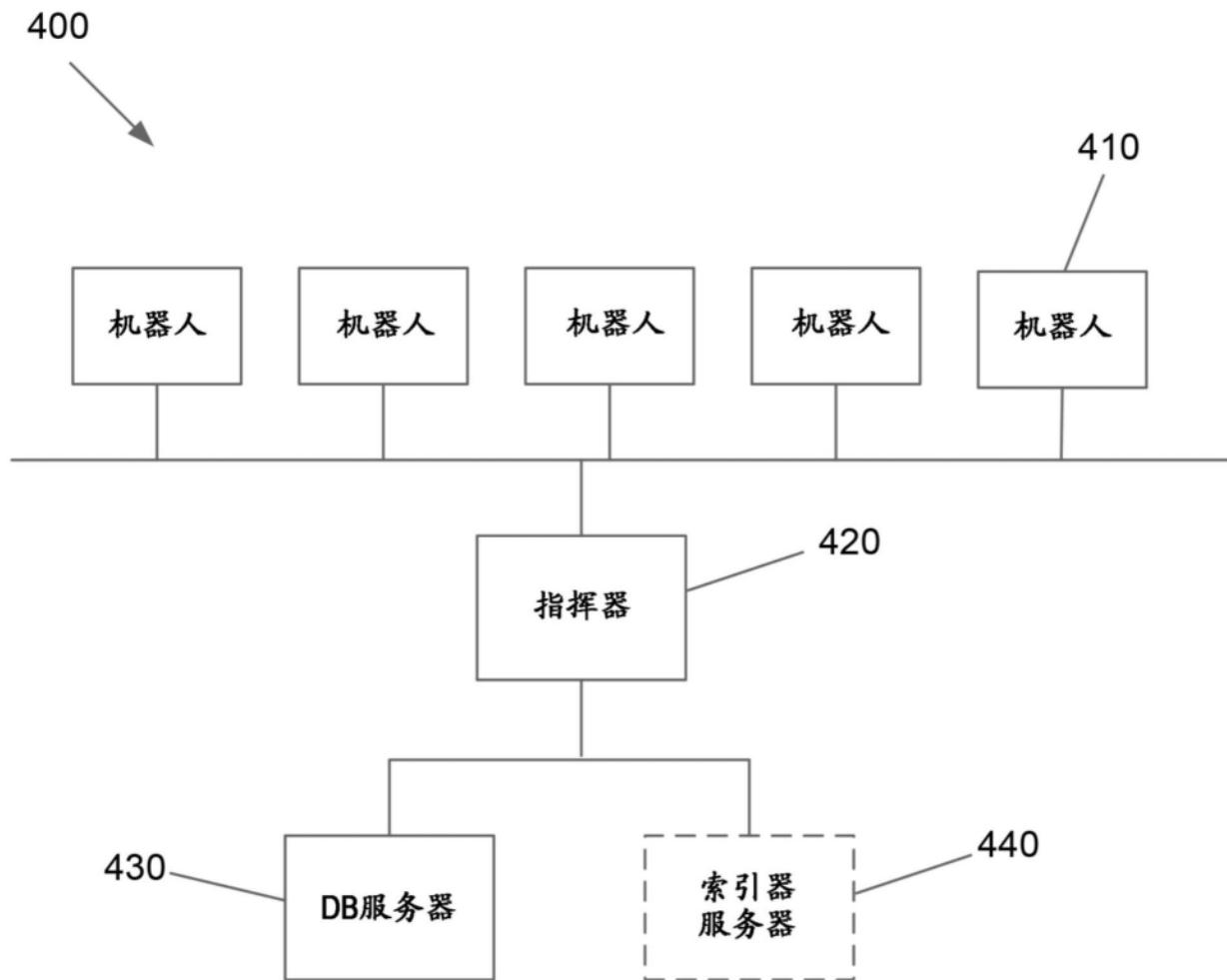


图4

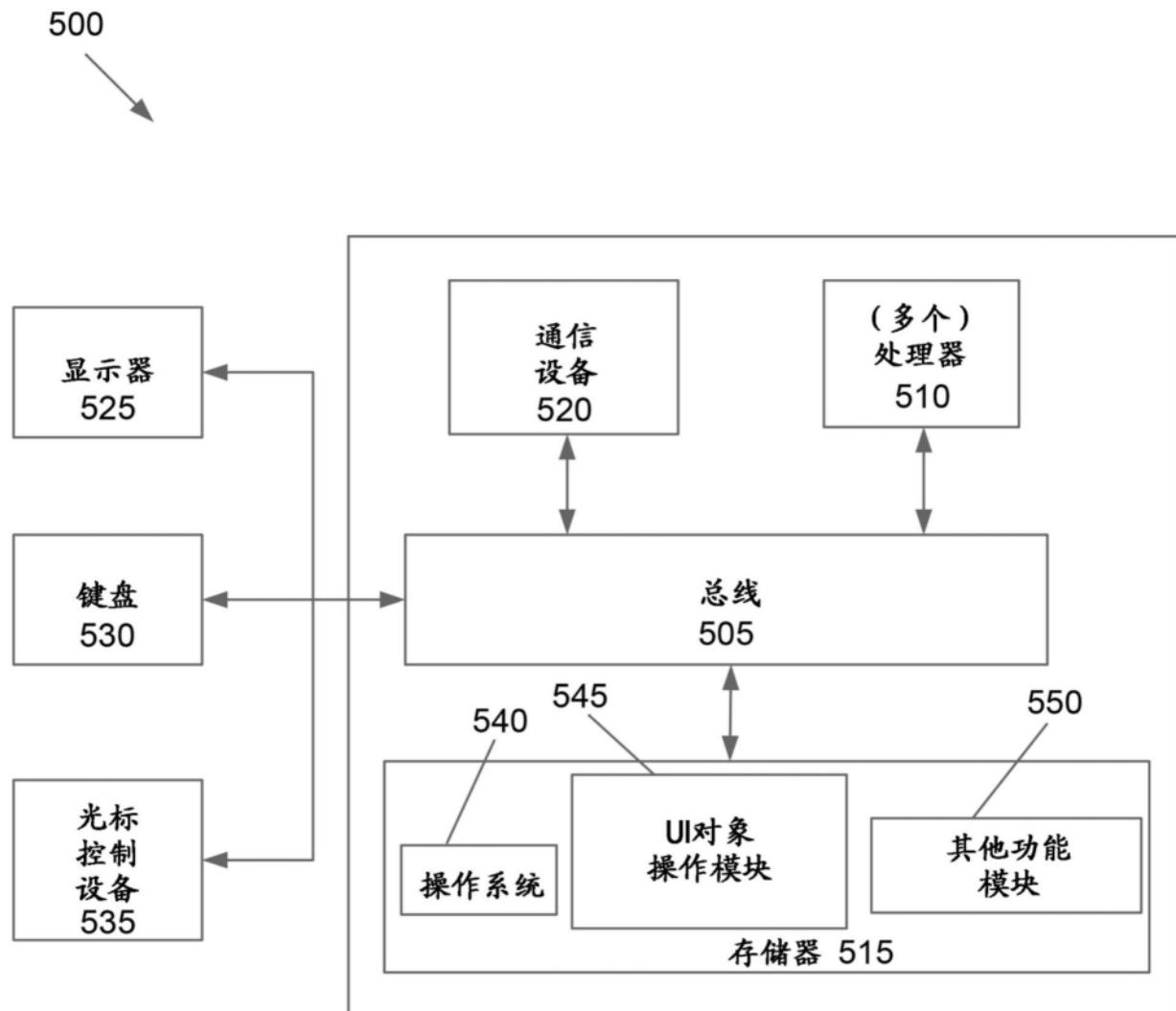


图5

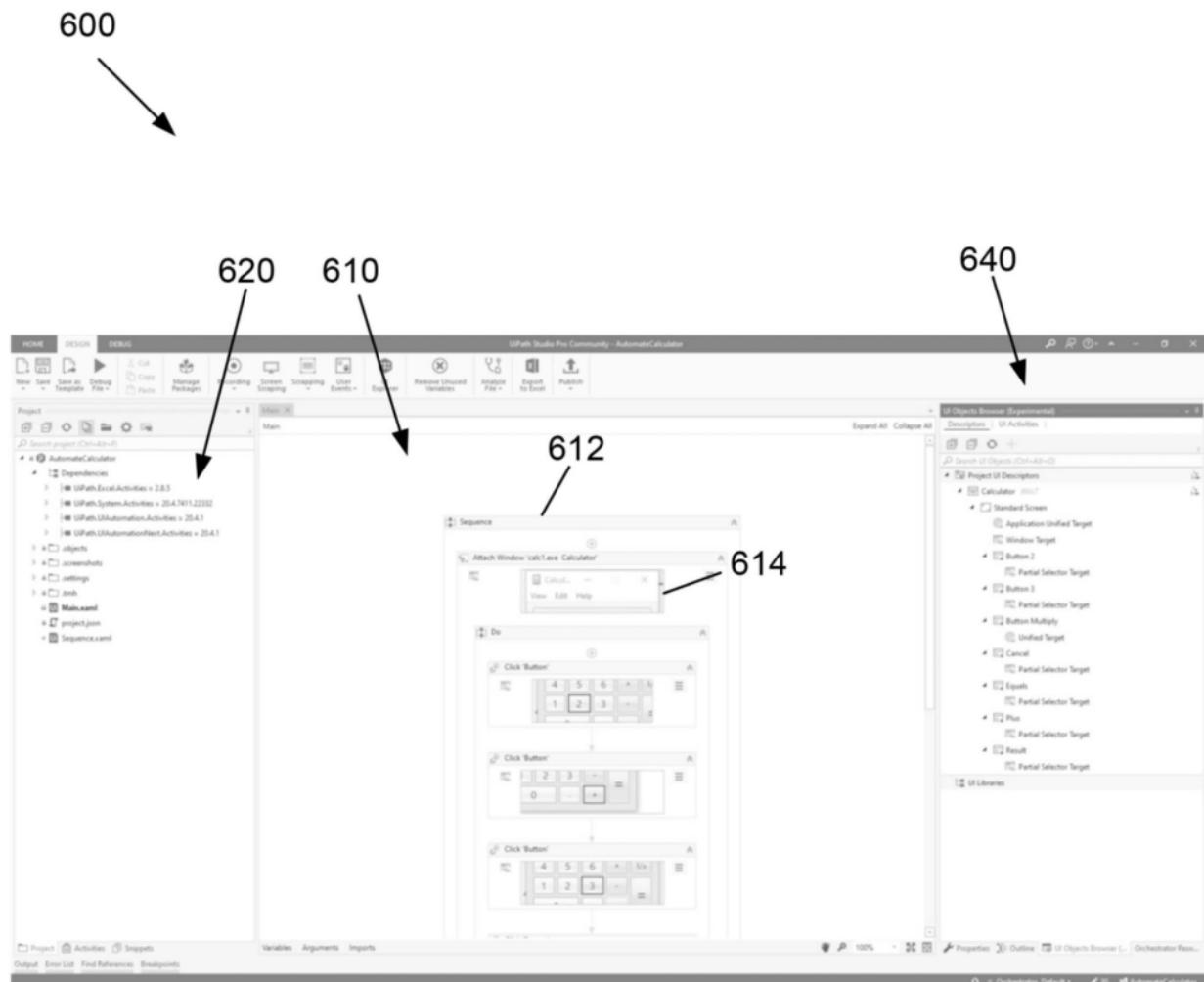


图6A



图6B

630



图6C

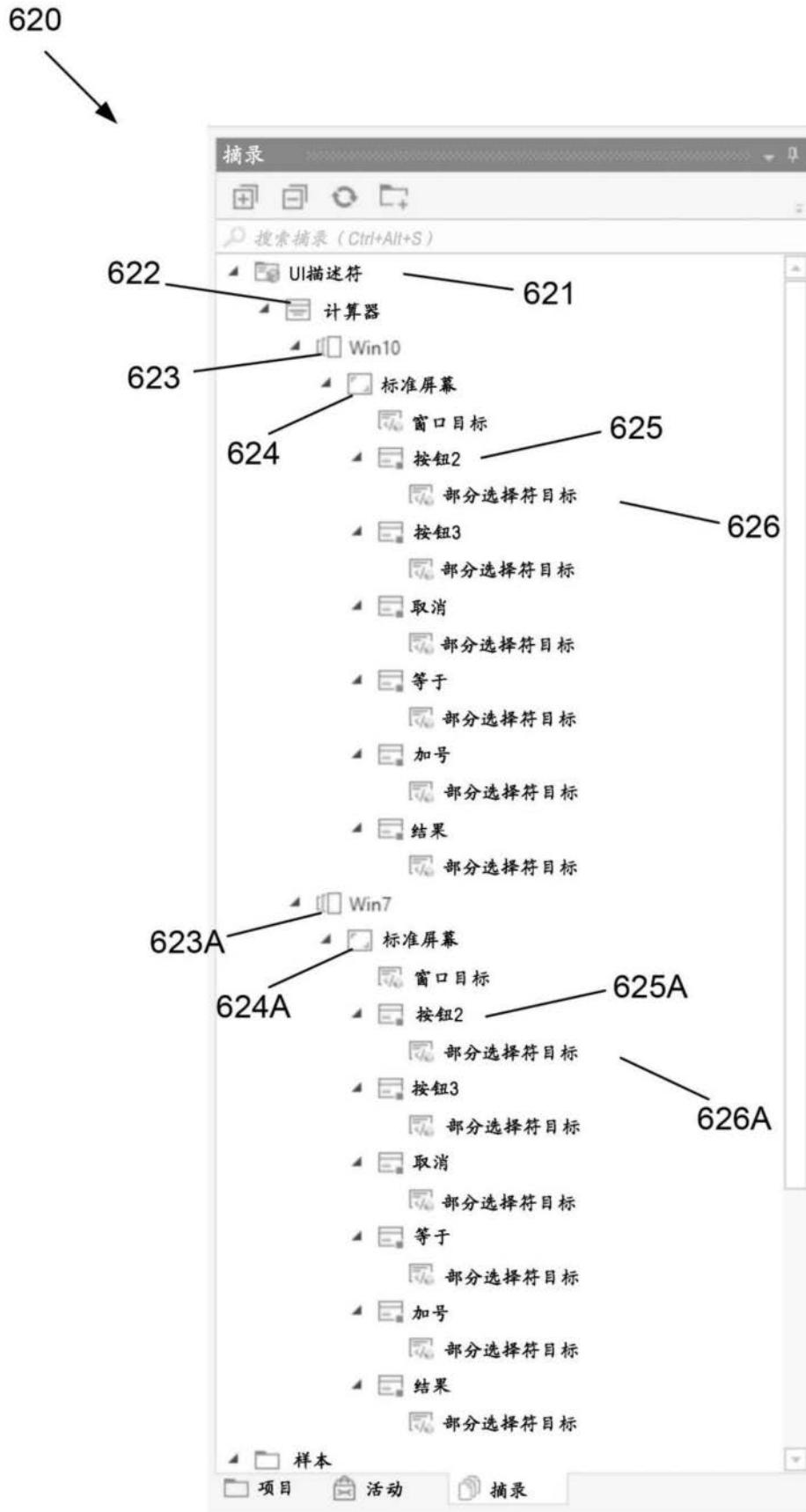


图6D

612

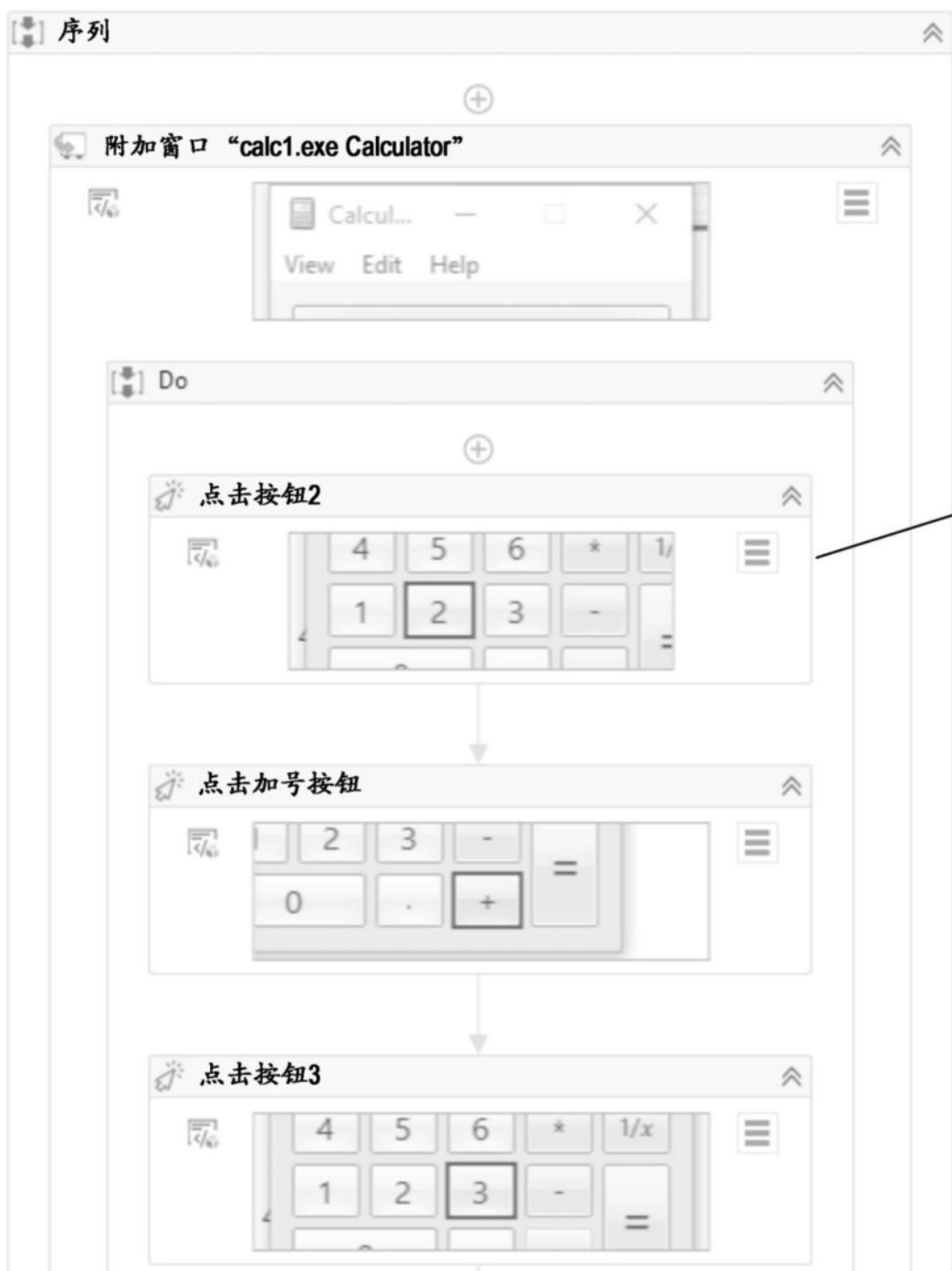


图6E

612

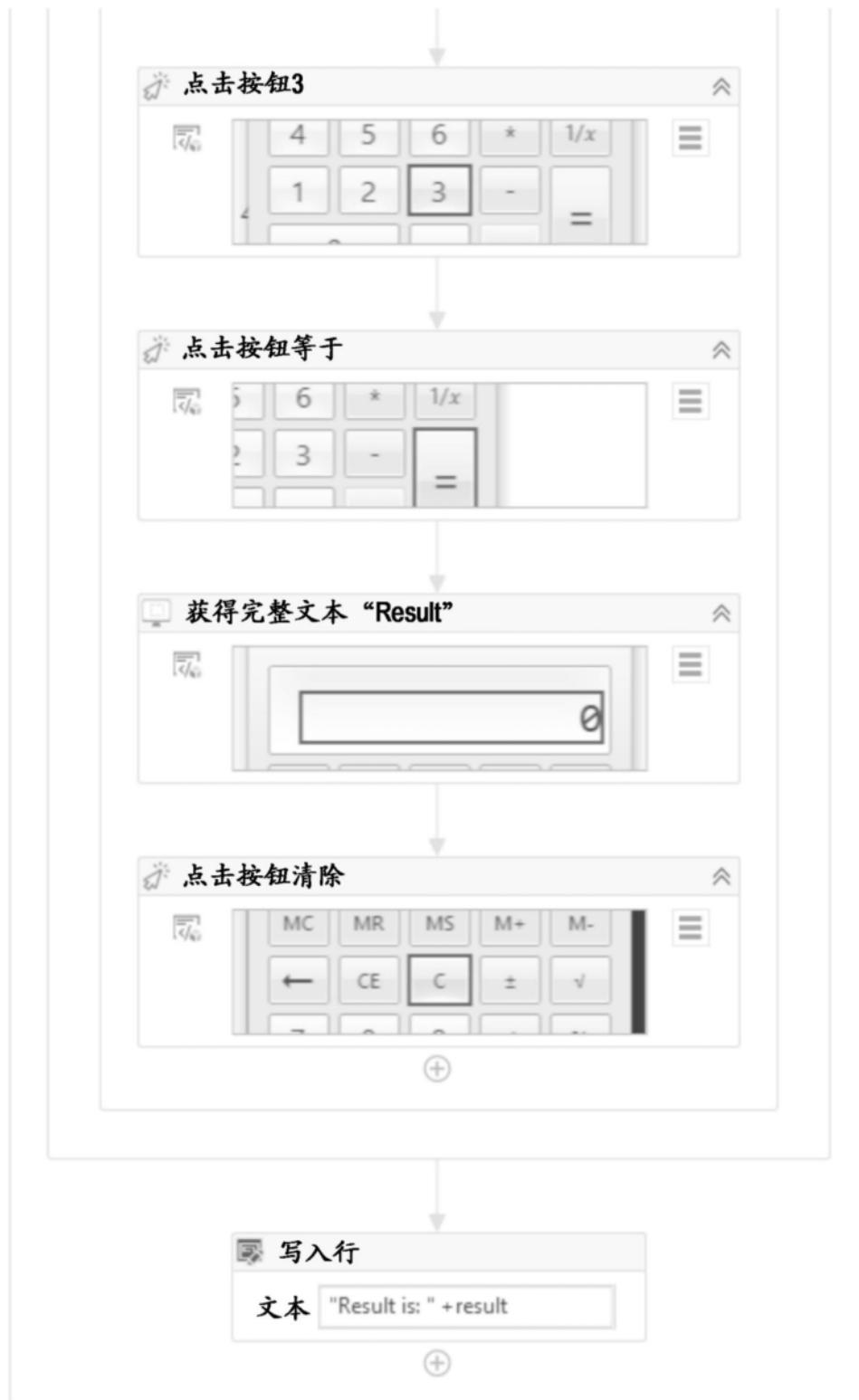


图6F



图6G

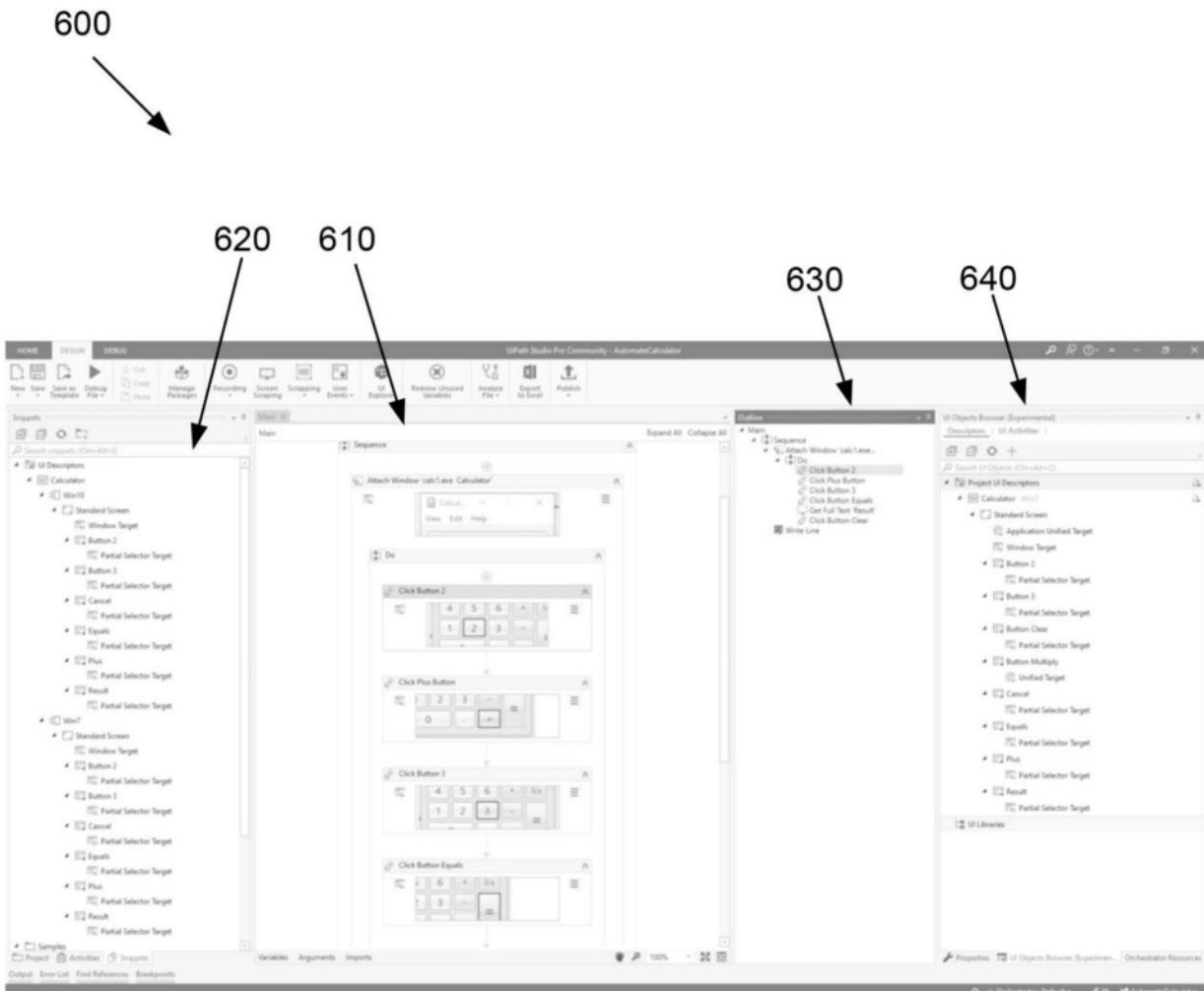
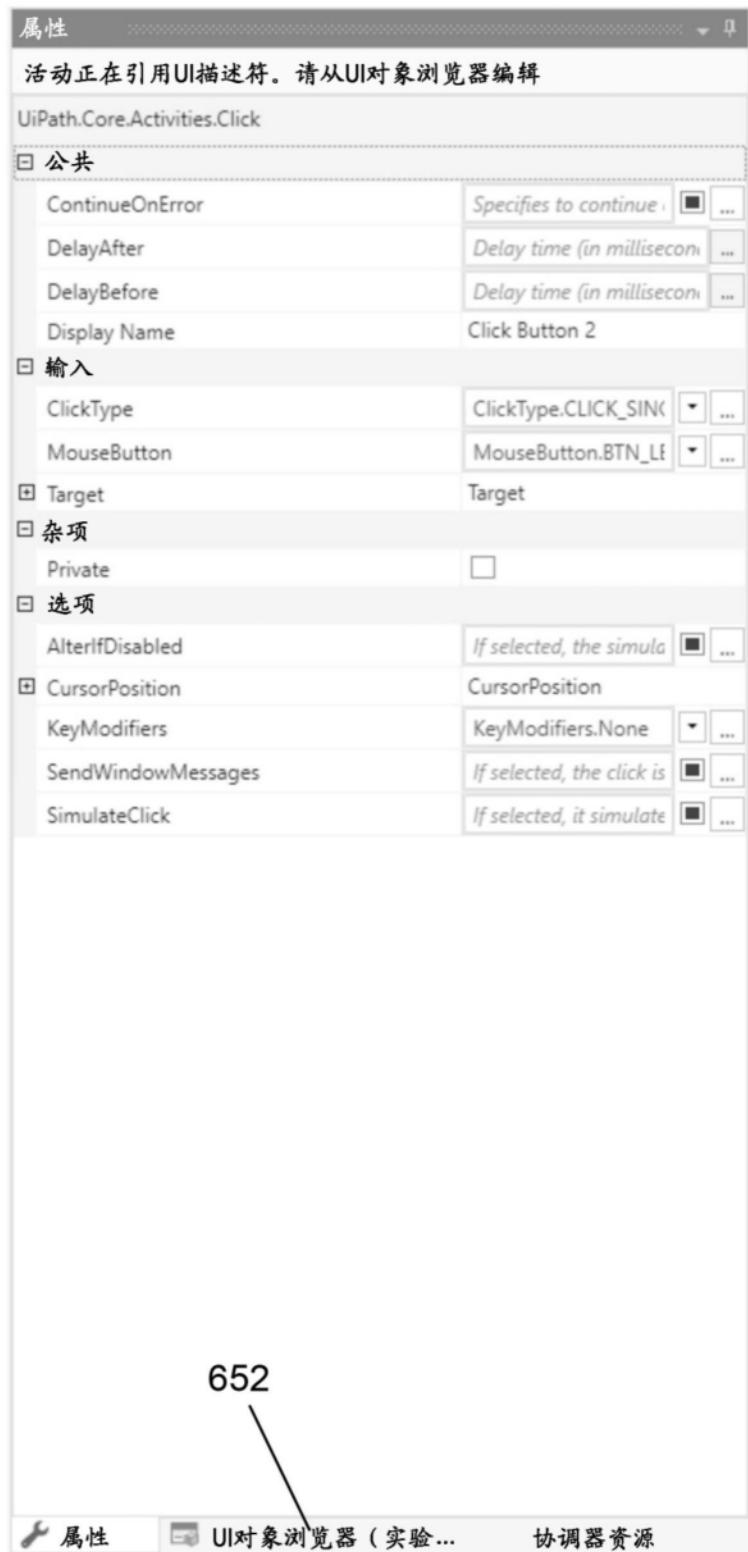


图6H

640



652

图6I

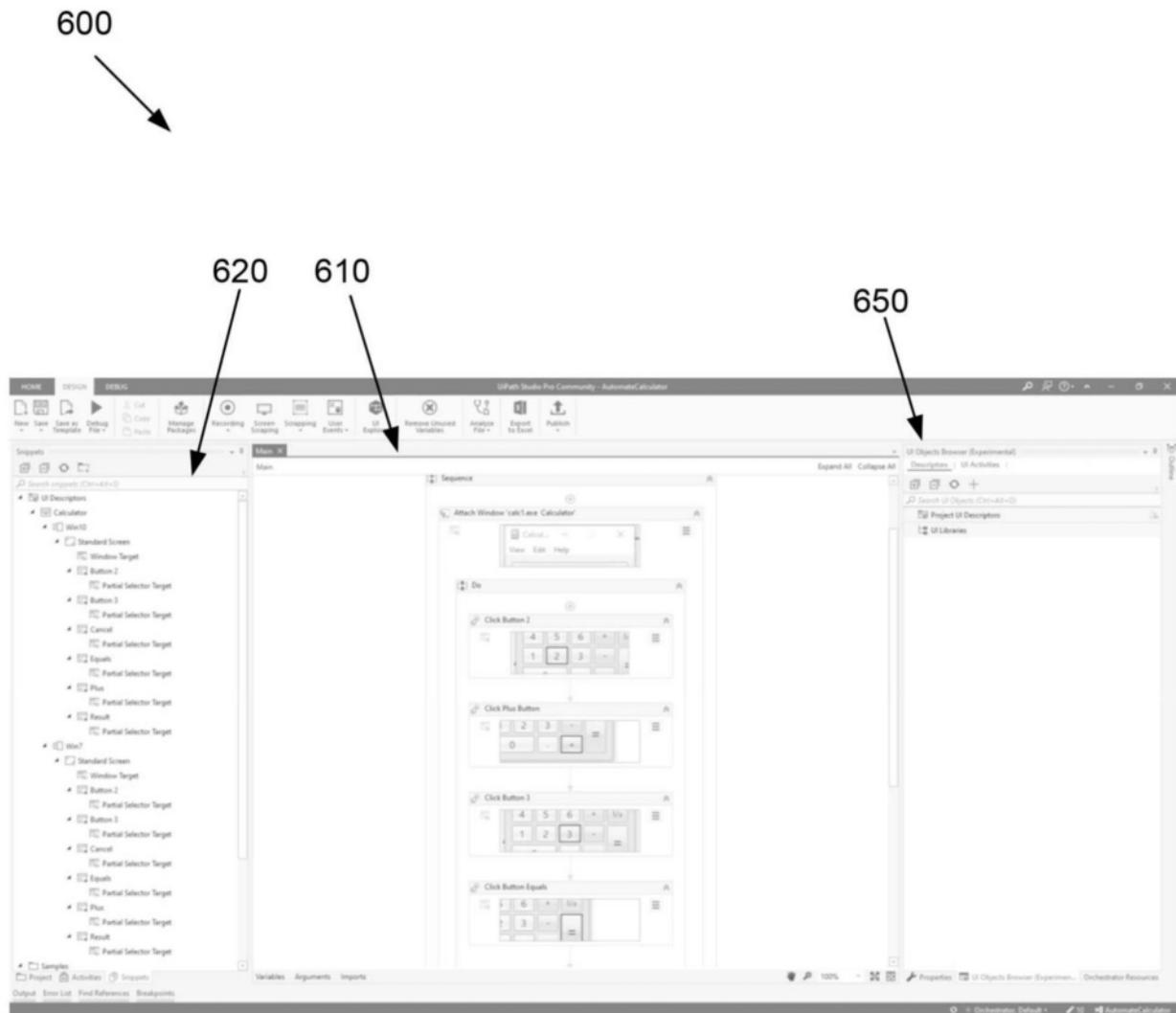


图6J



图6K

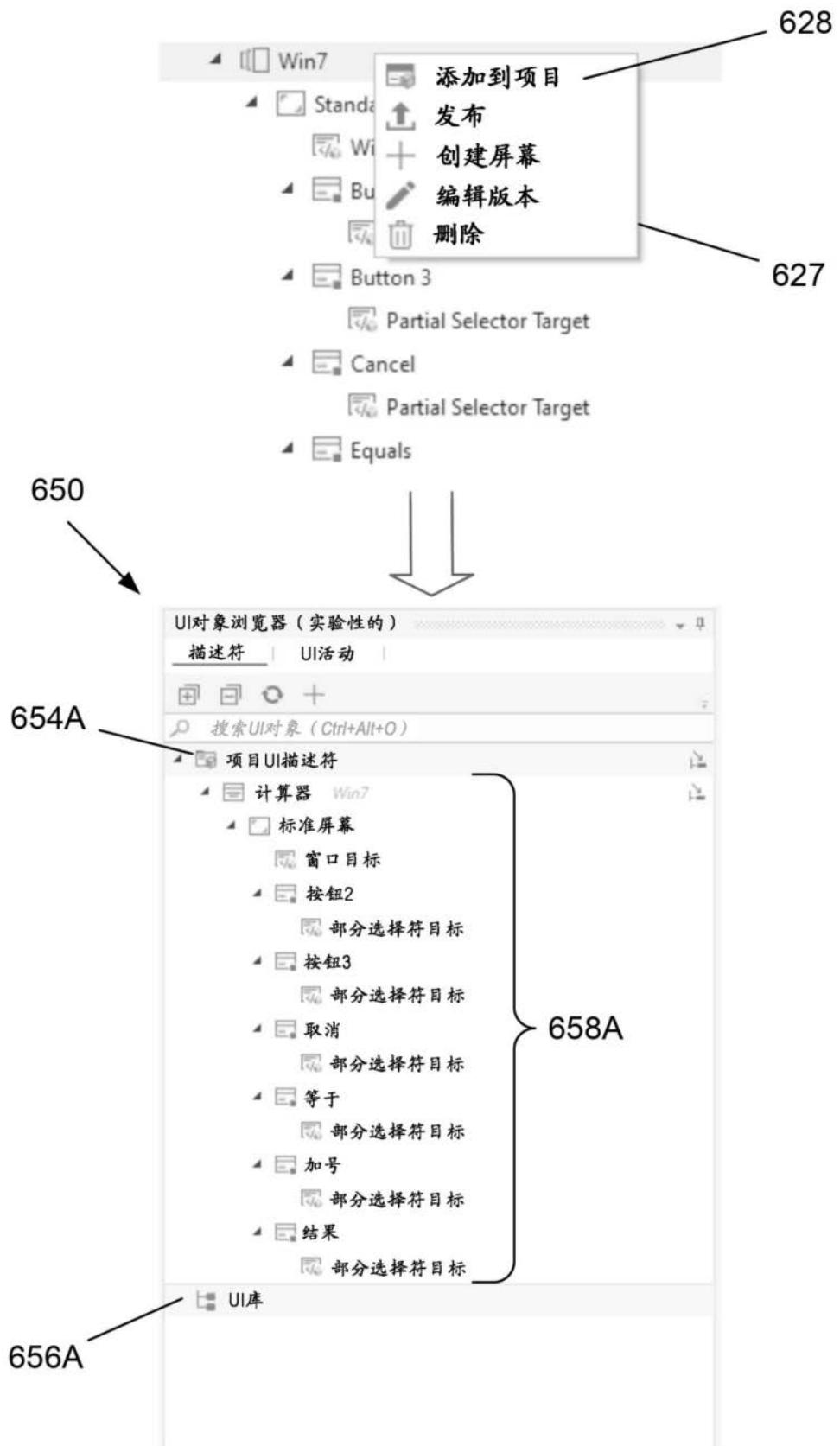


图6L

650



图6M

612

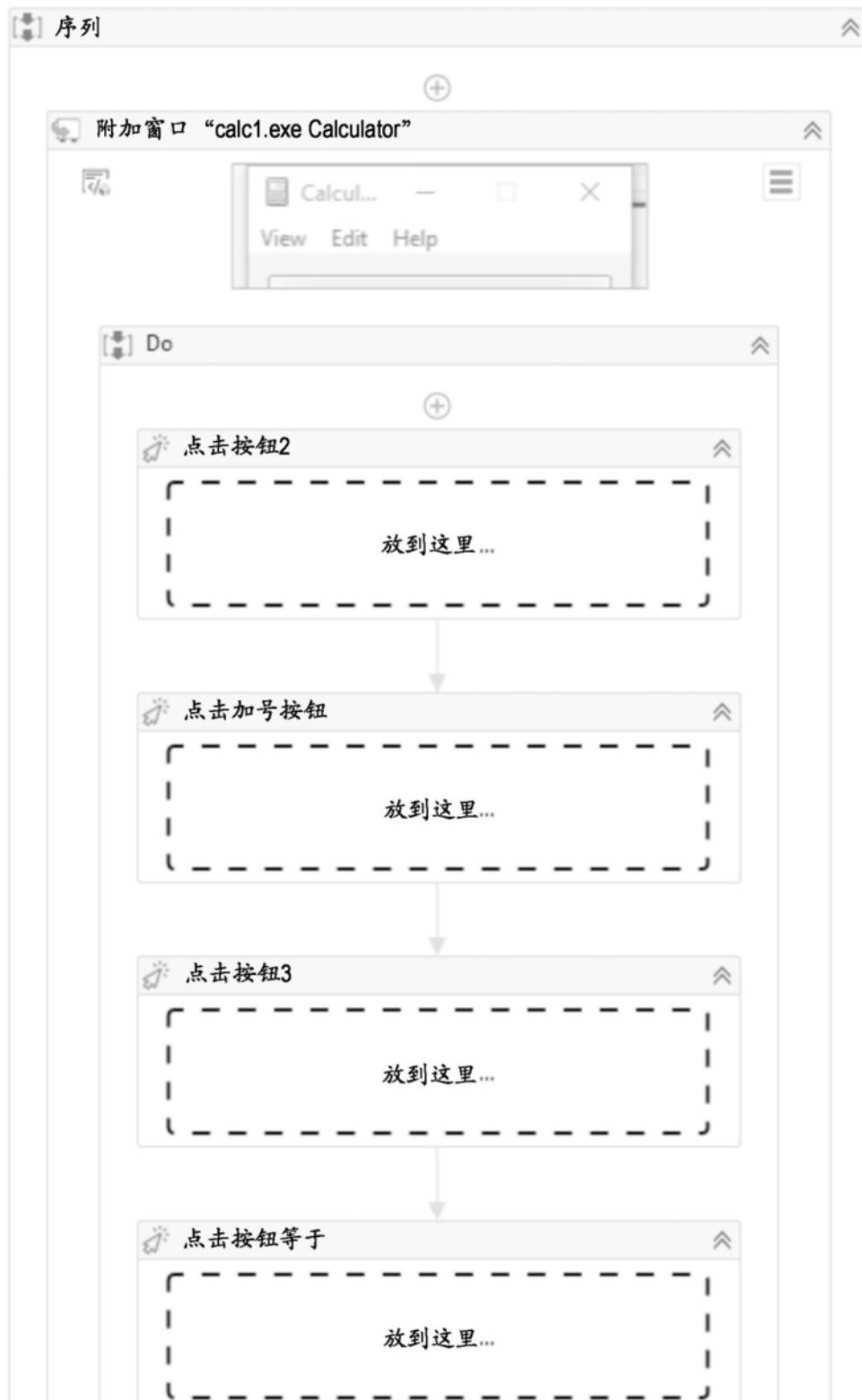


图6N



图60

650

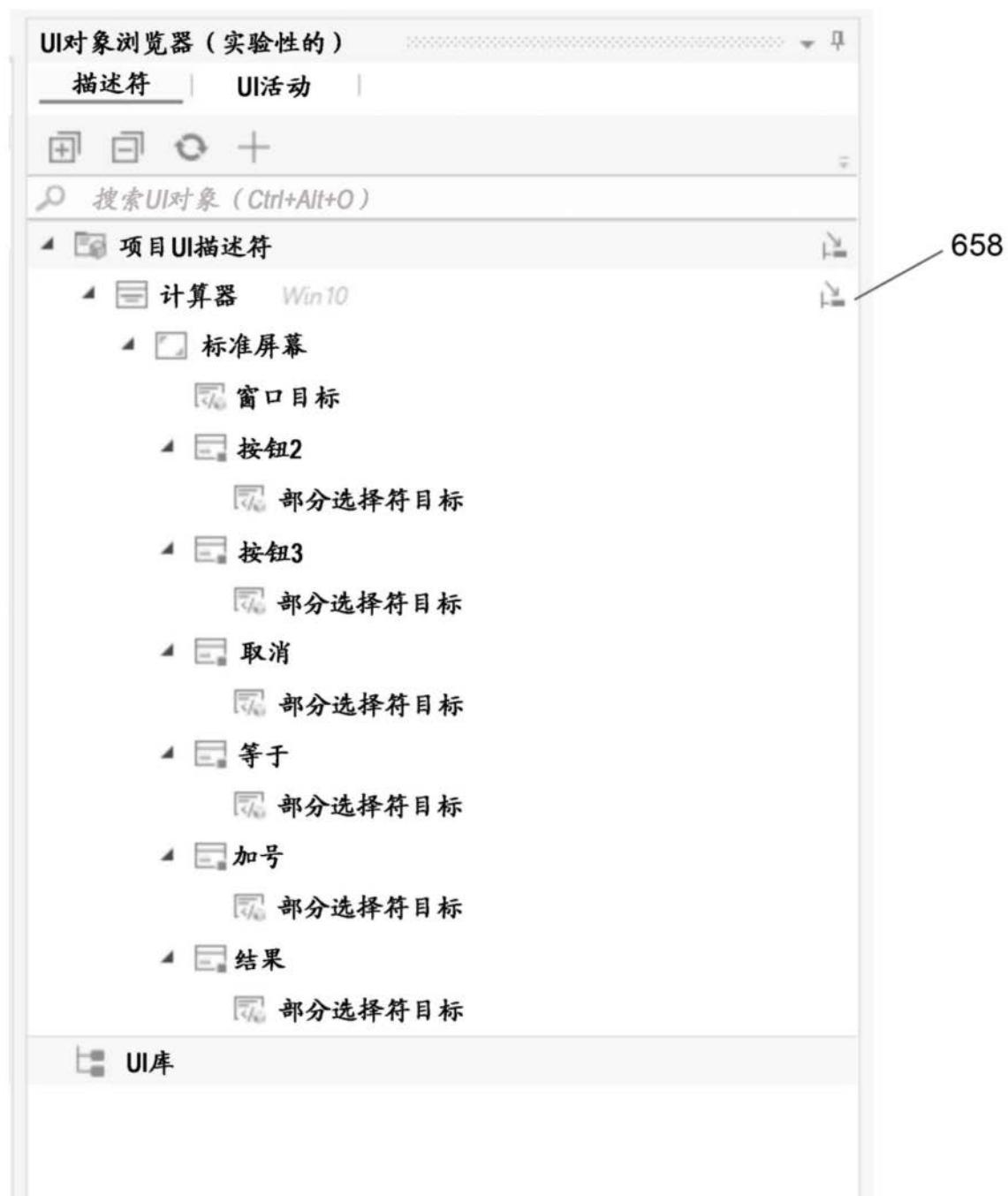


图6P

650

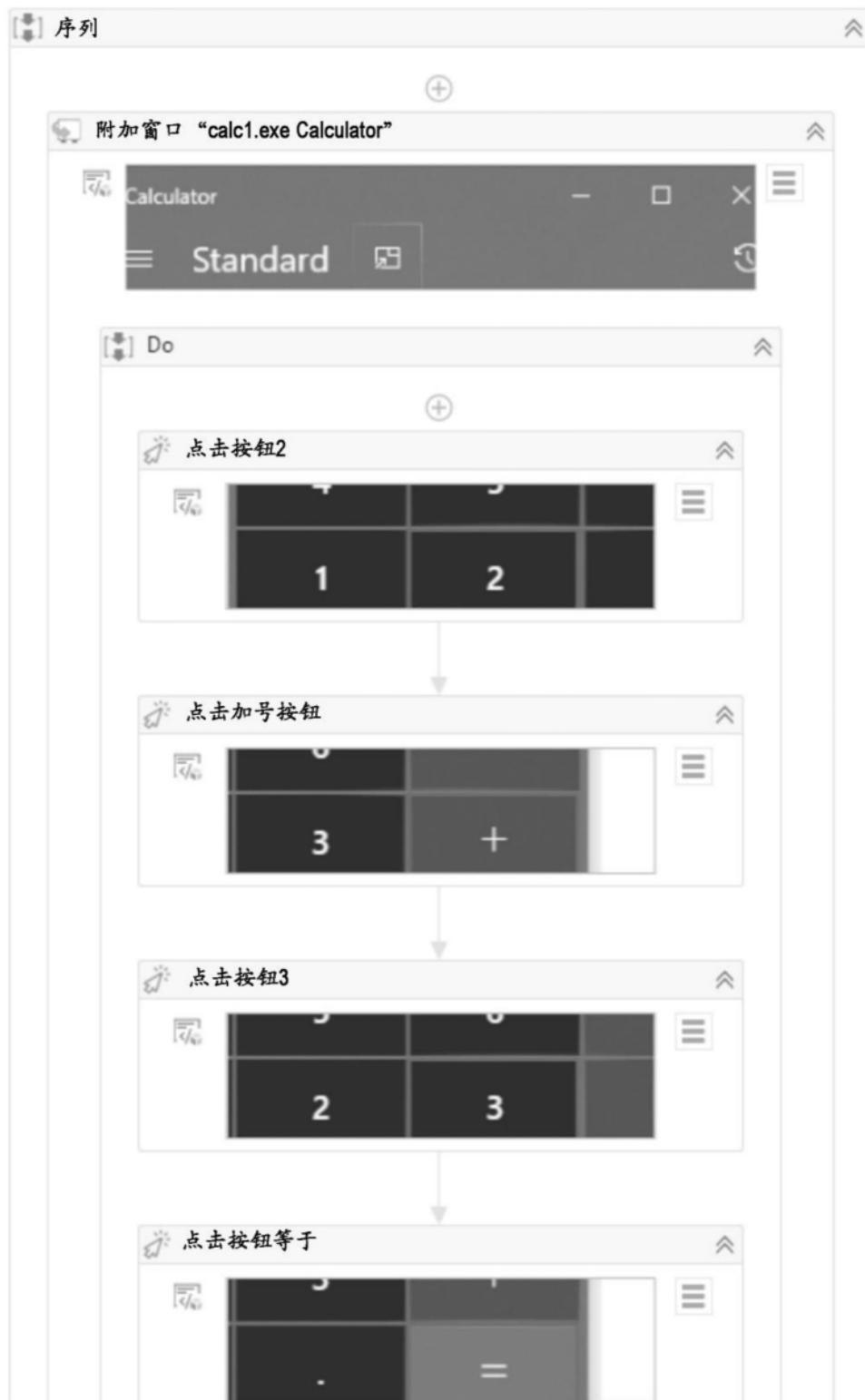


图6Q



图6R



图6S

650

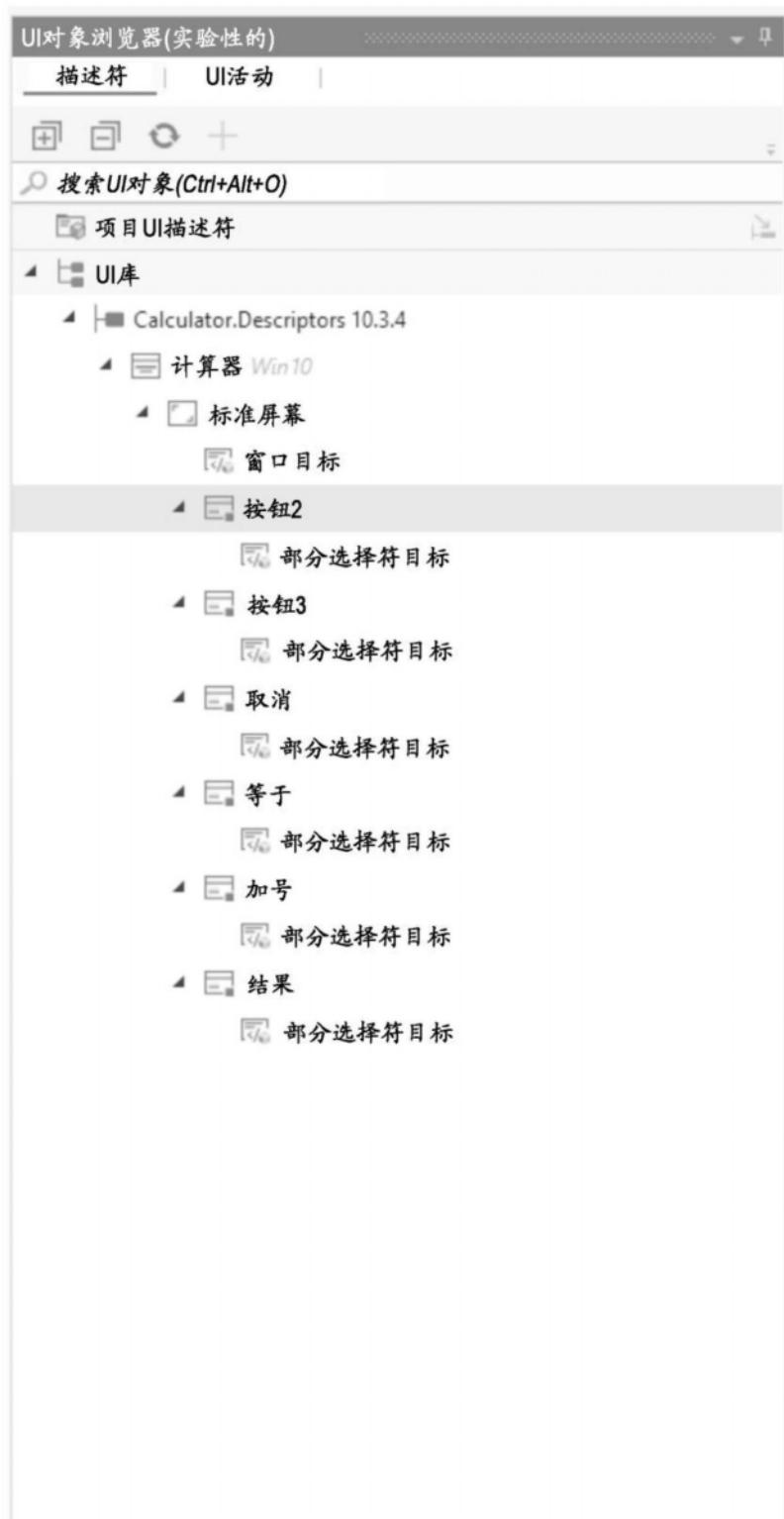


图6T

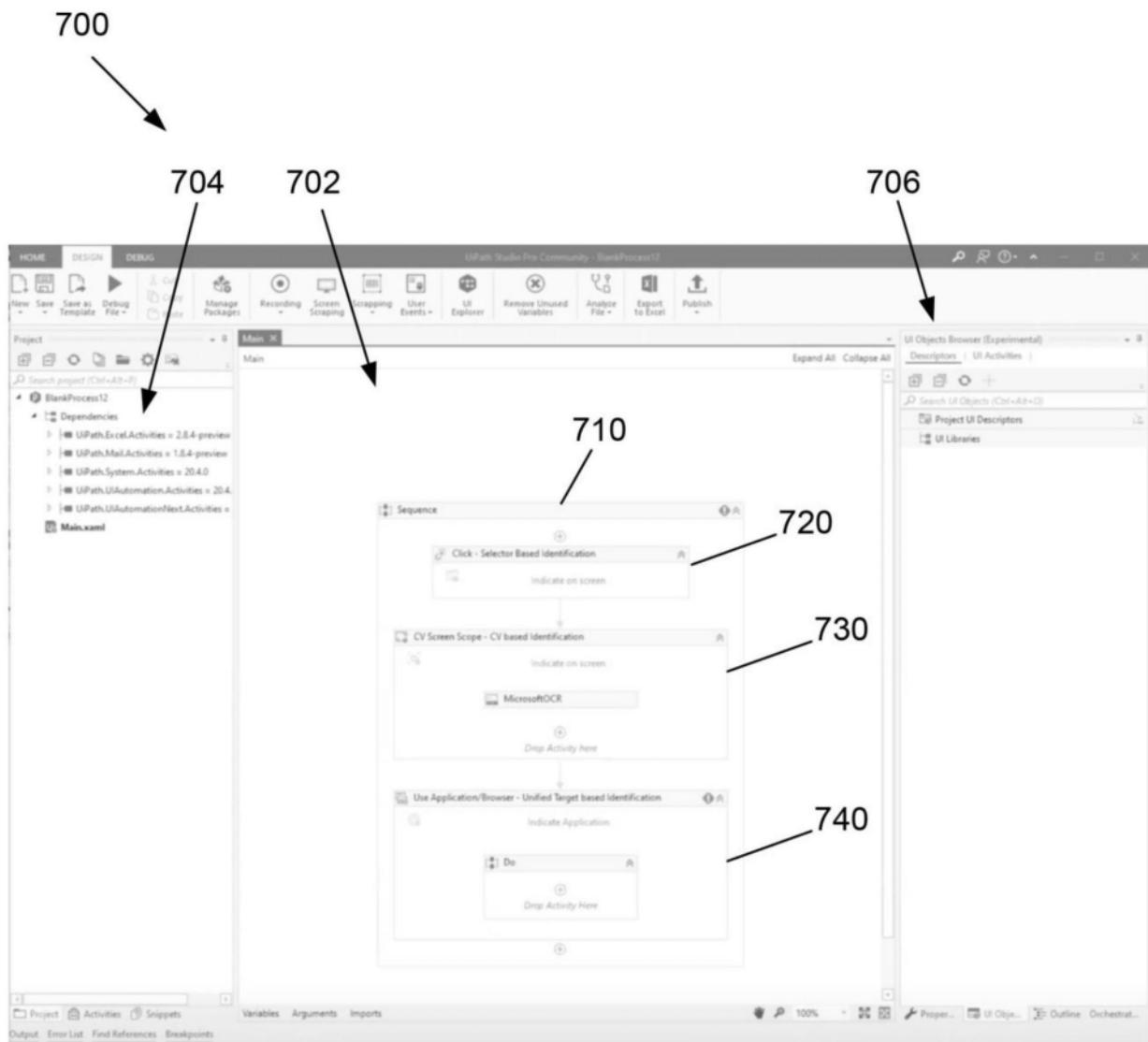


图7A

710

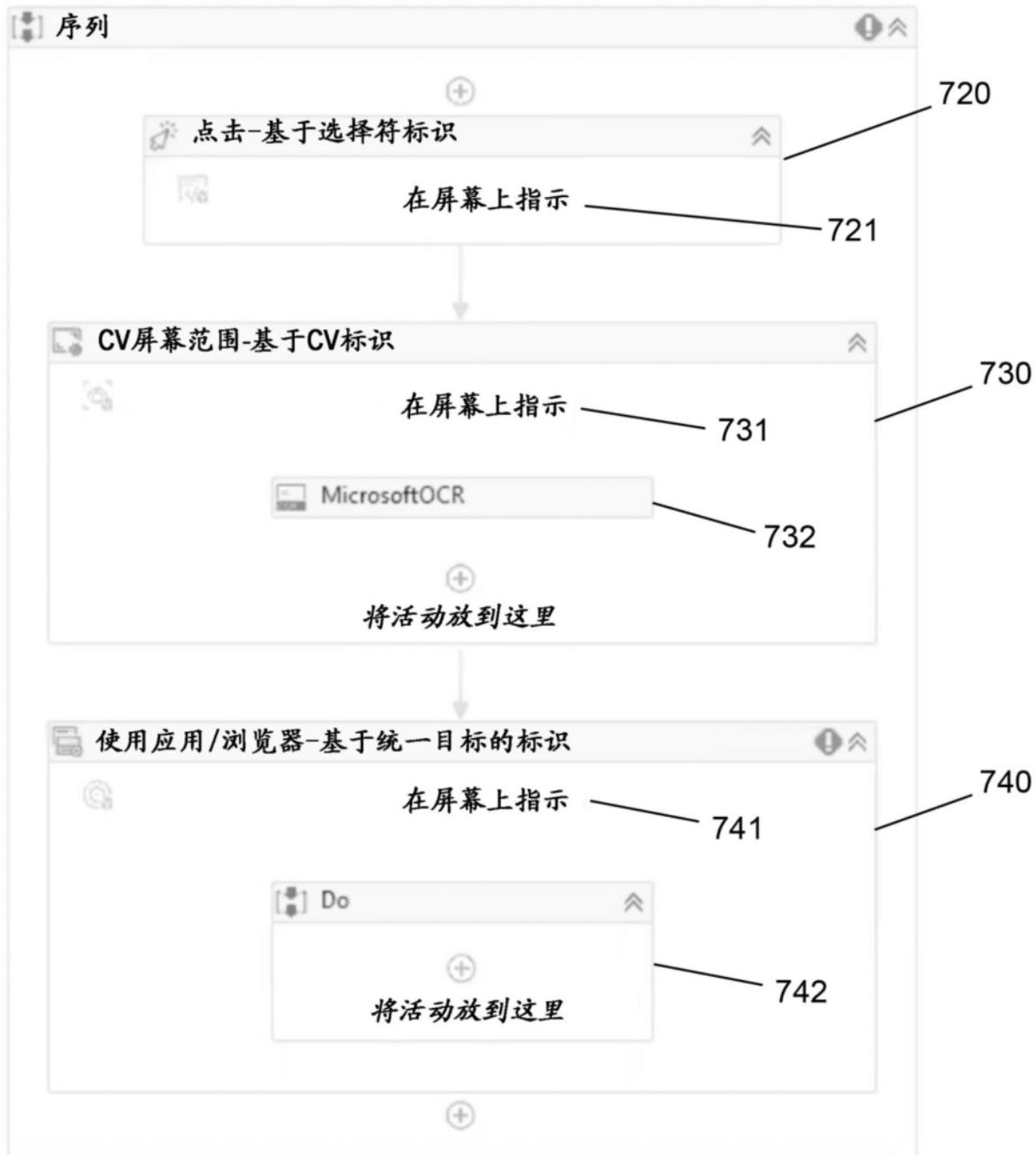


图7B

750



图7C

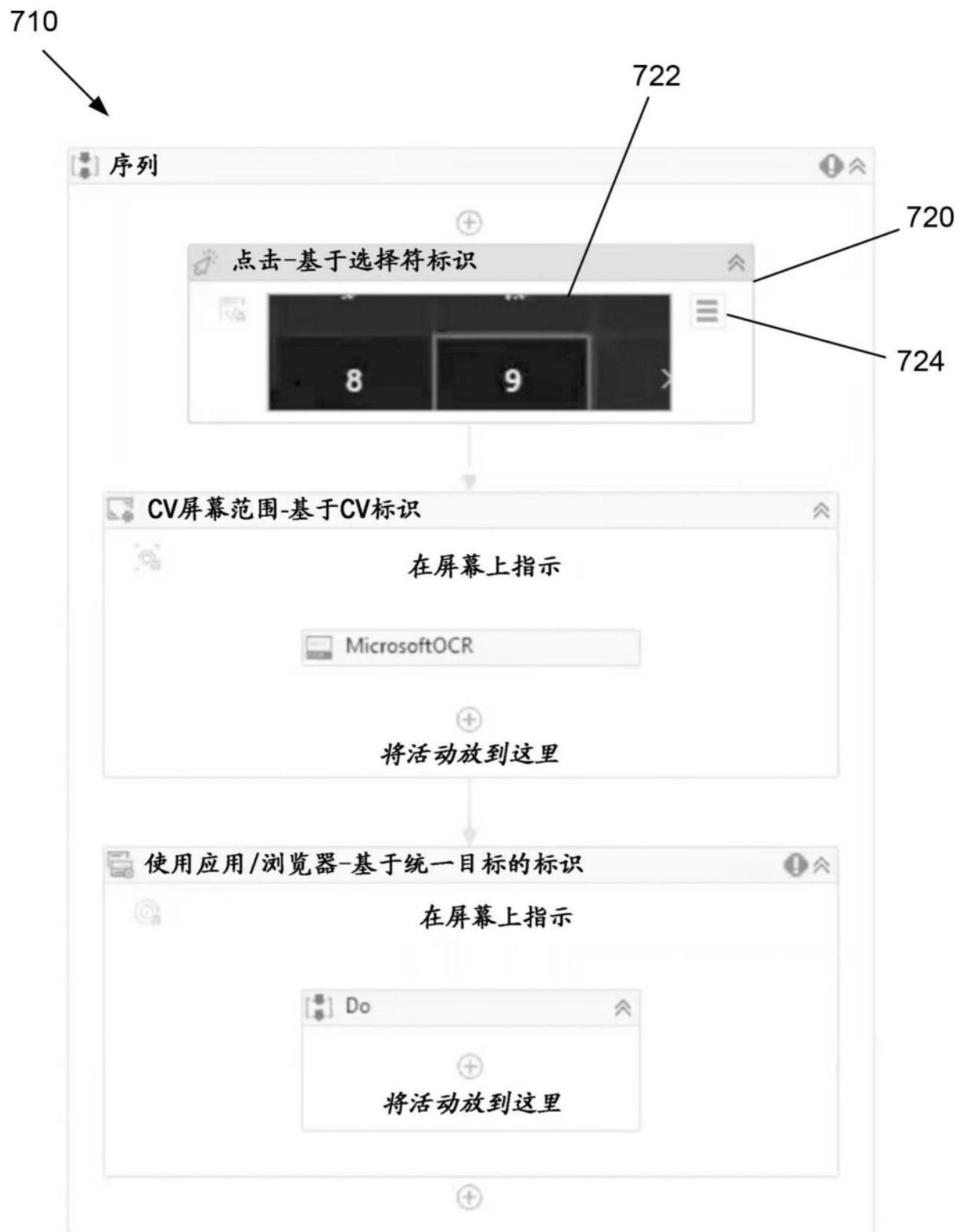


图7D

725



图7E



图7F

728



图7G

706



图7H

710



图7I



图7J

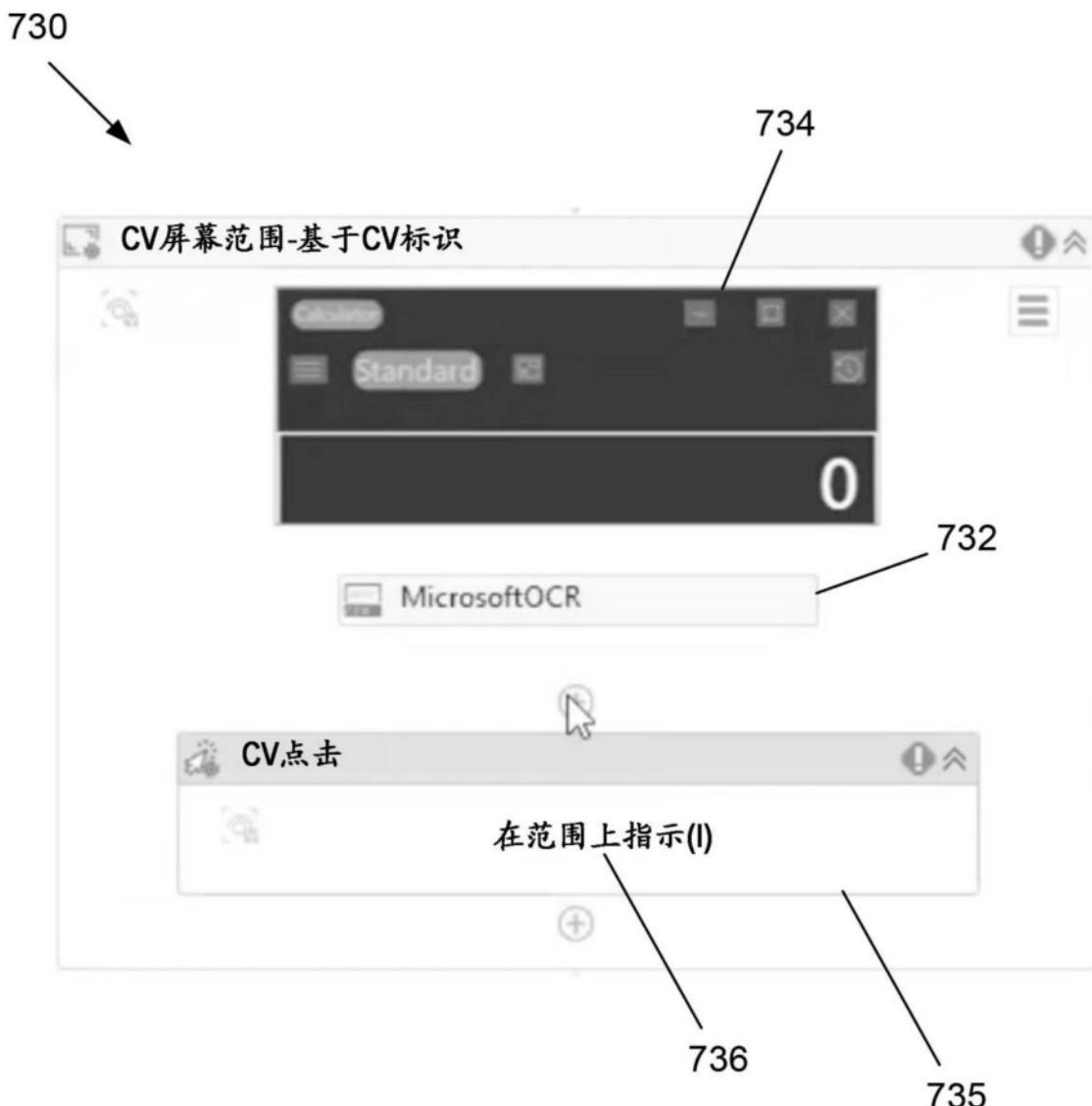


图7K

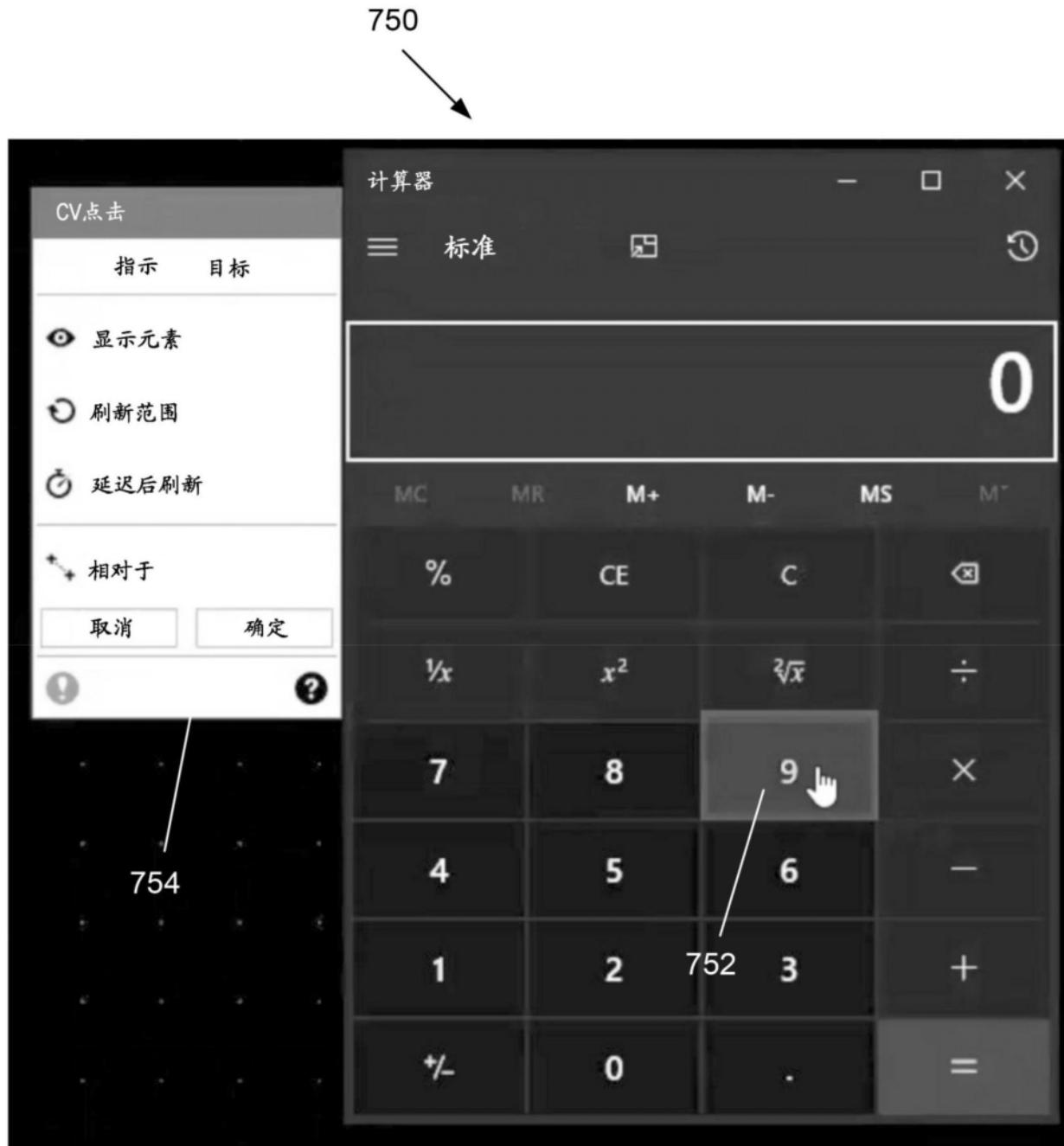


图7L



图7M

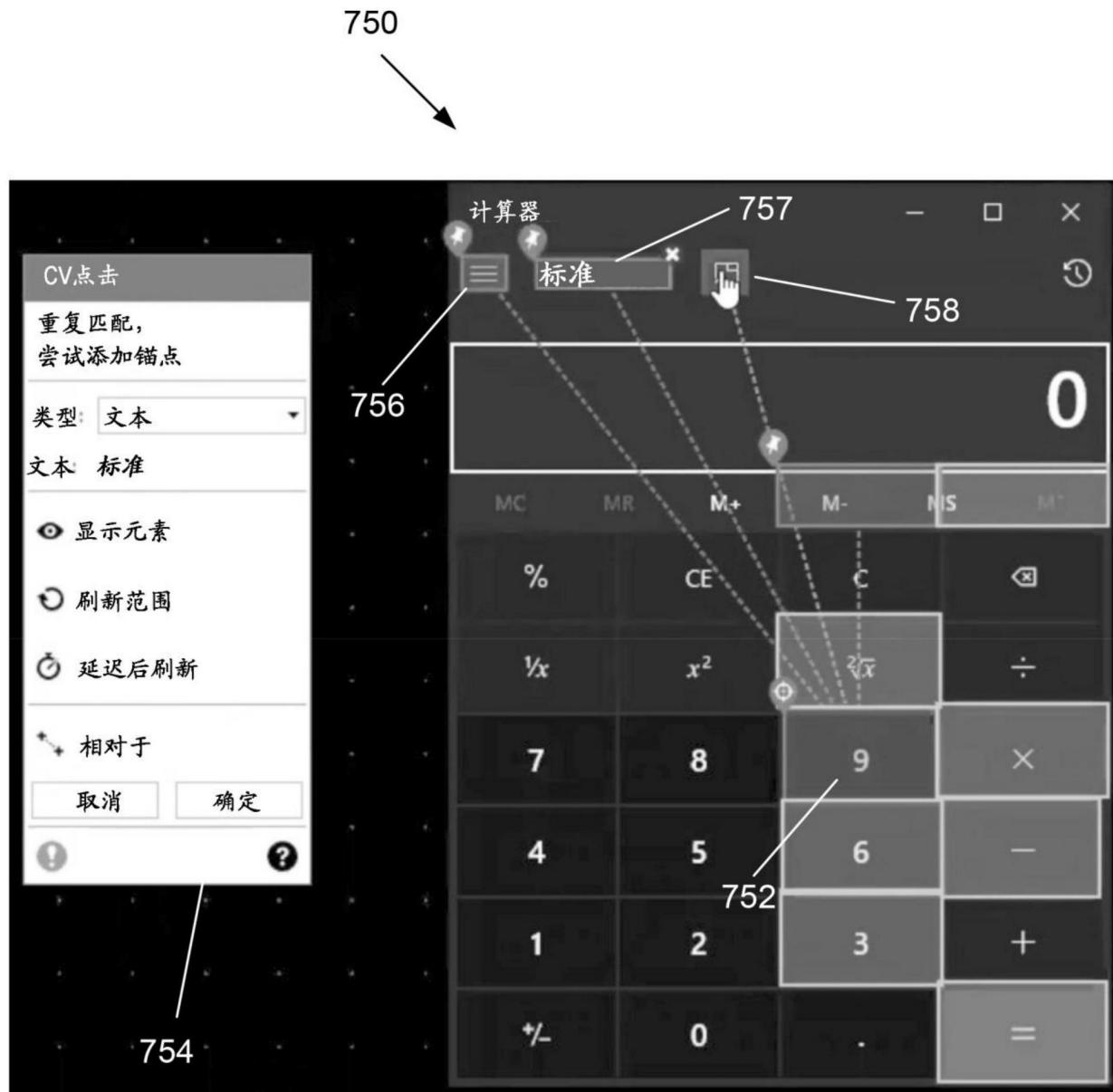


图7N

710

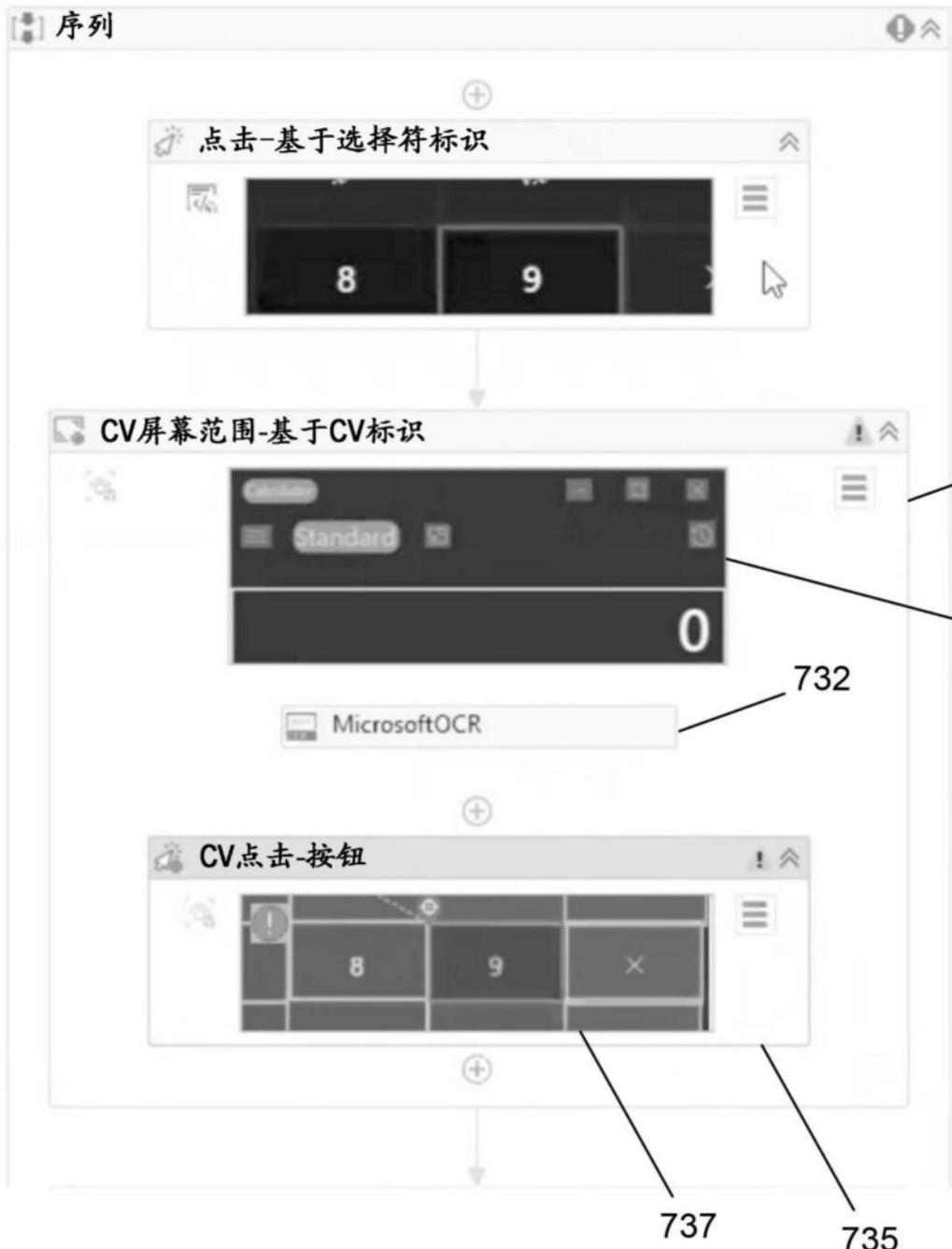


图70

760

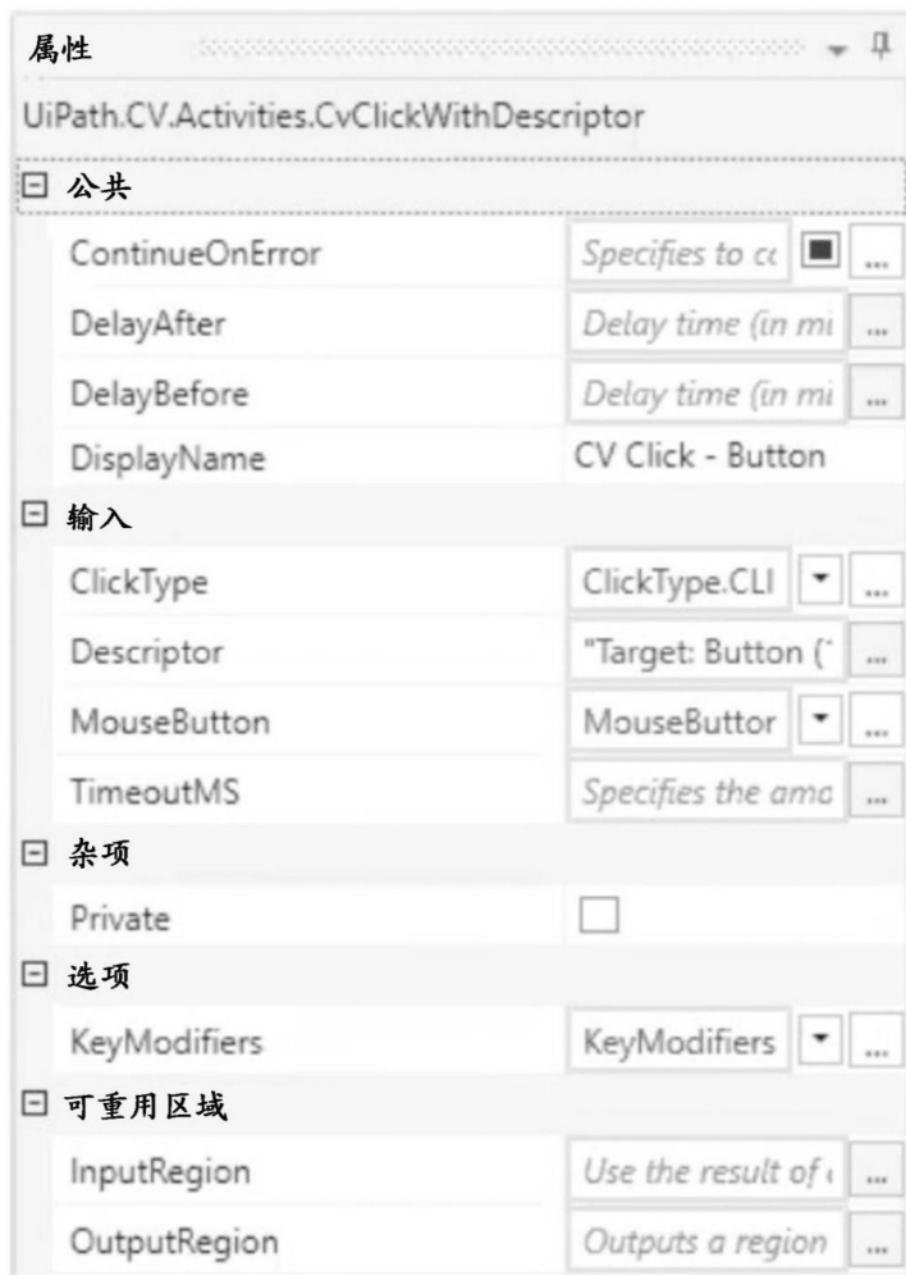


图7P

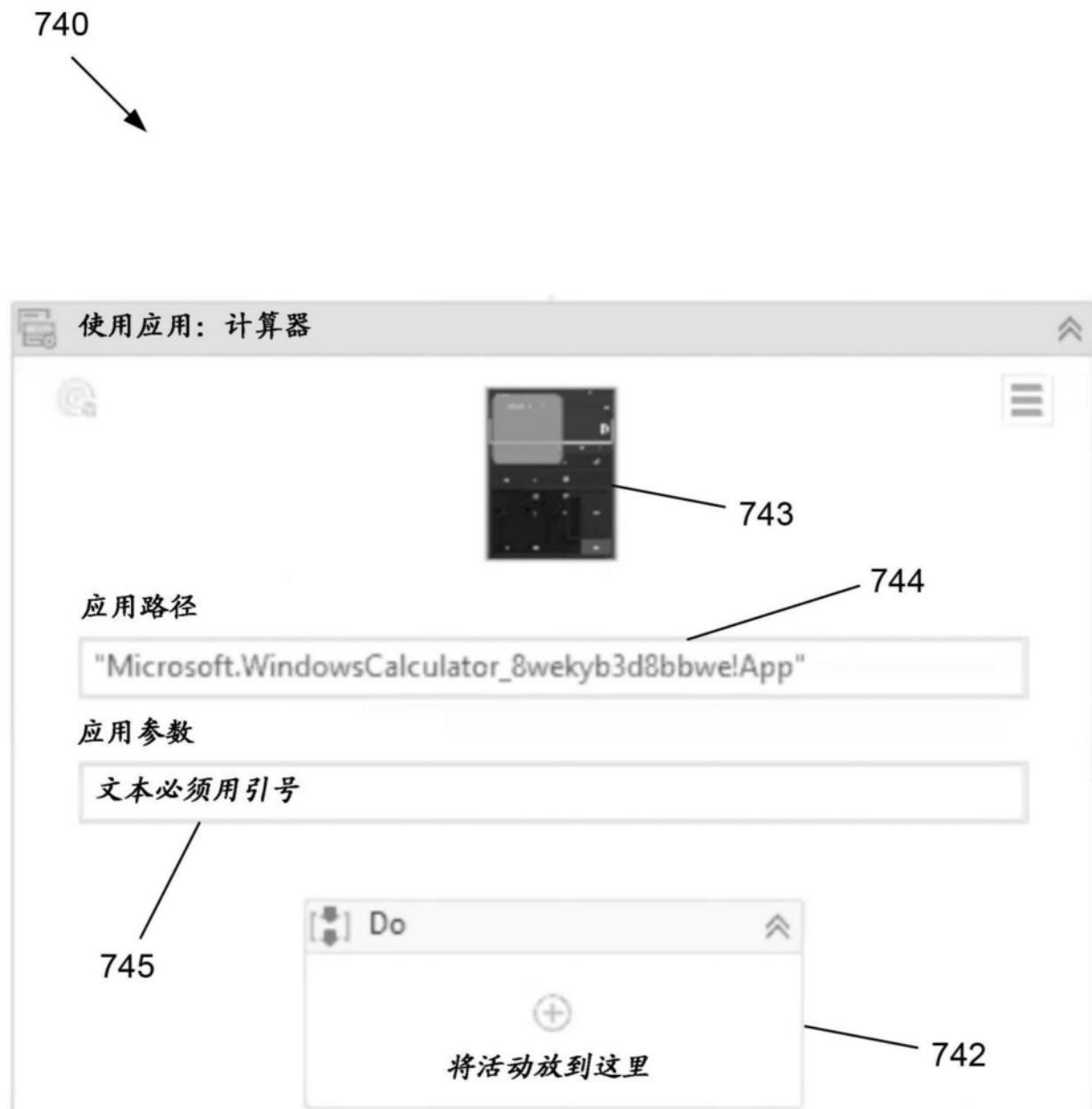


图7Q

708

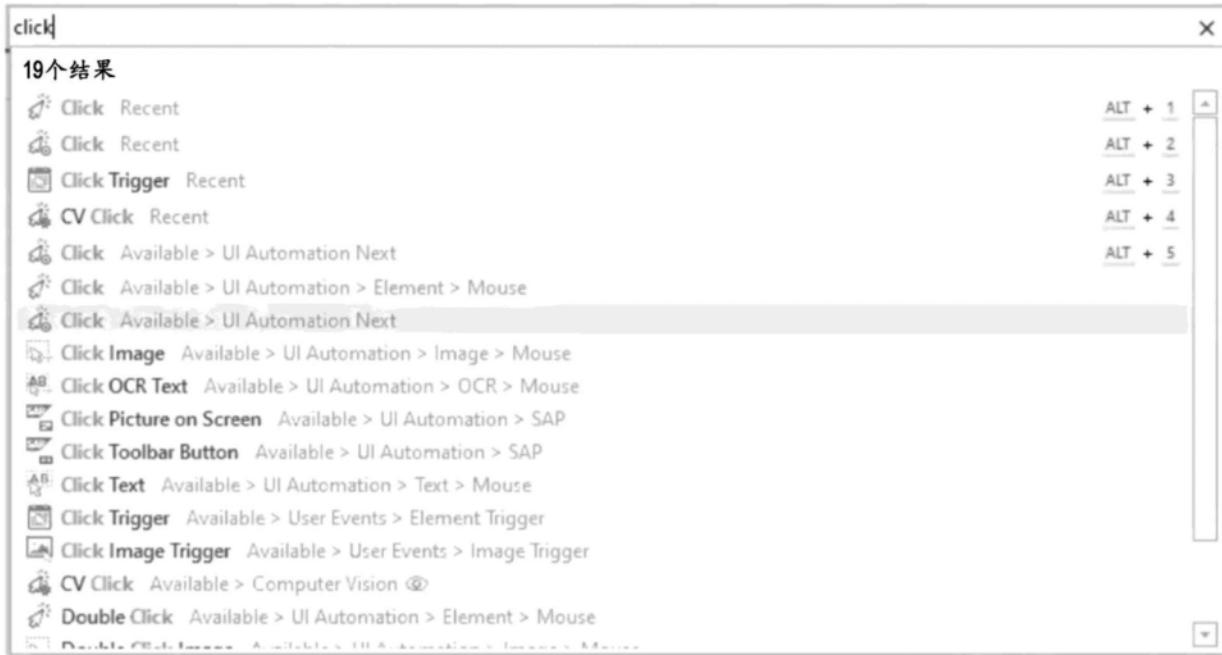


图7R

740

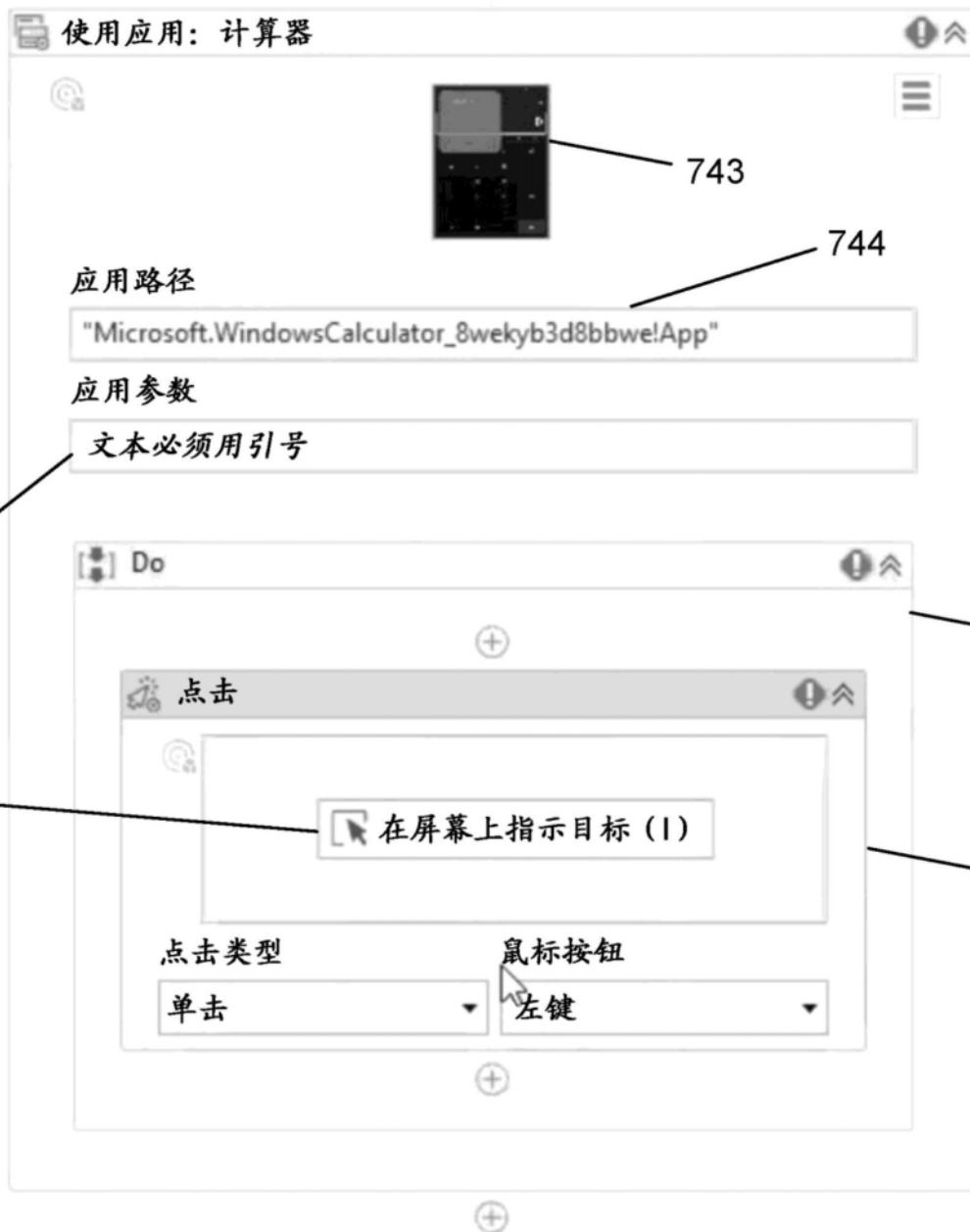


图7S



图7T

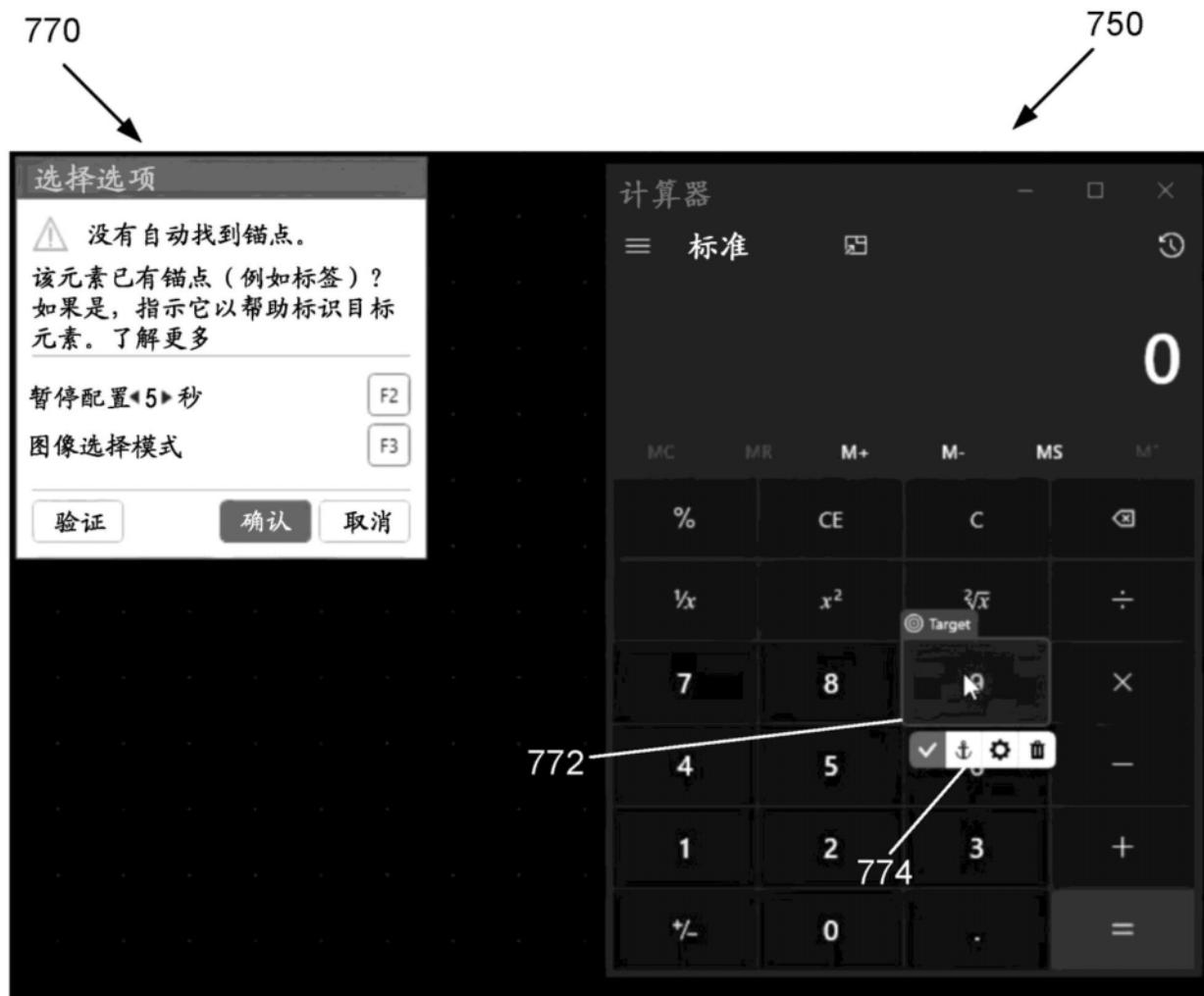


图7U

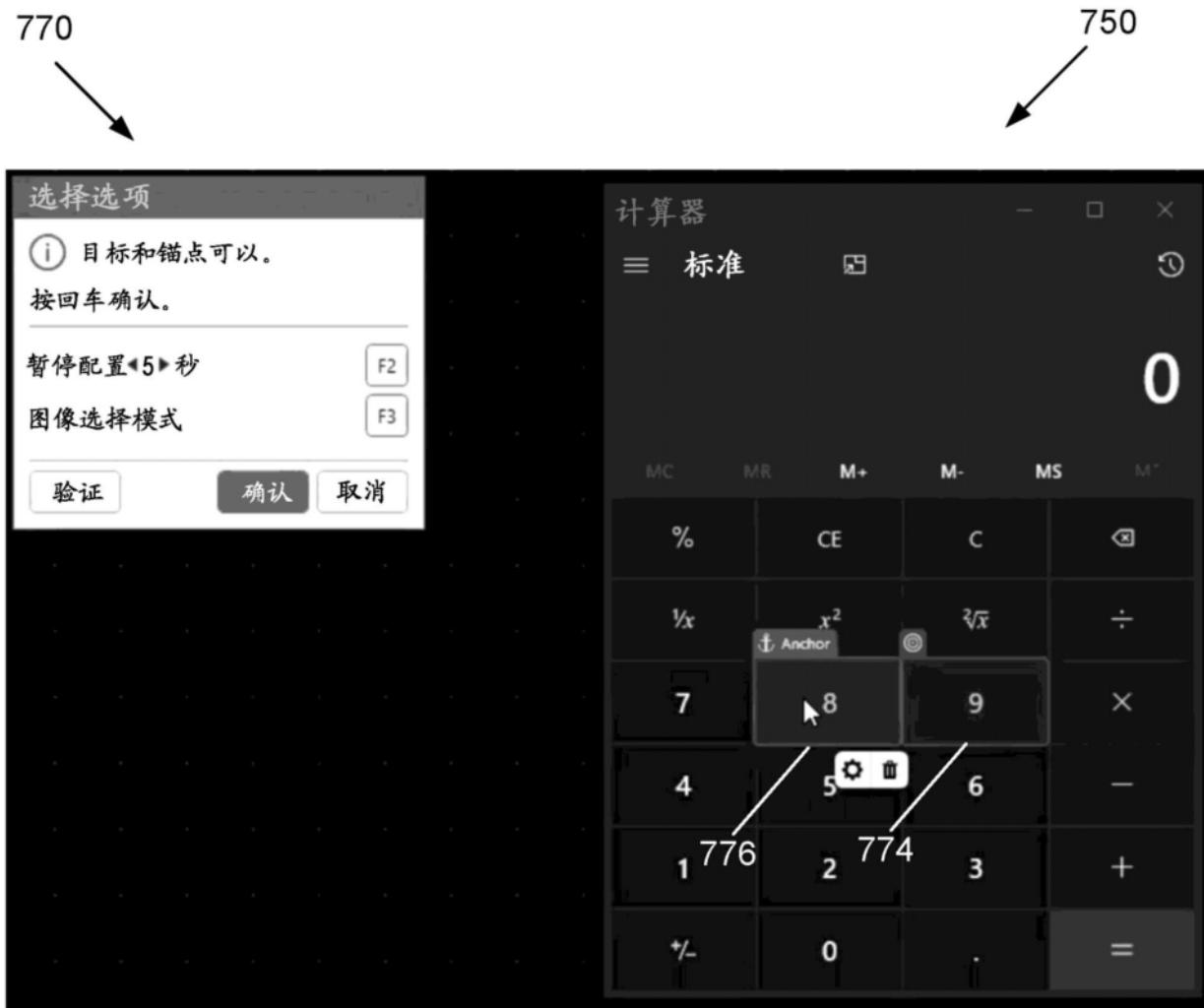


图7V



图7W

740



图7X

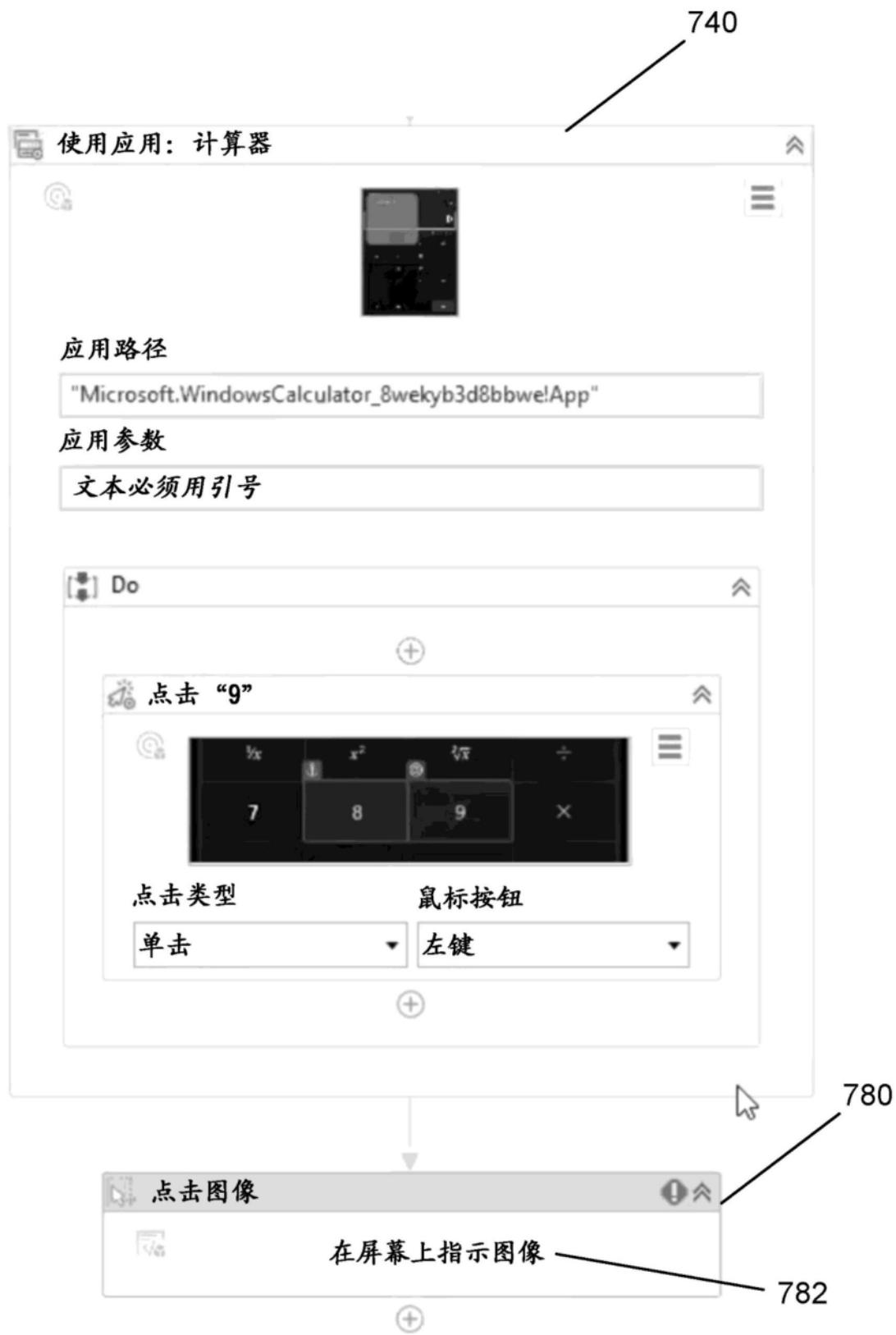


图7Y

750

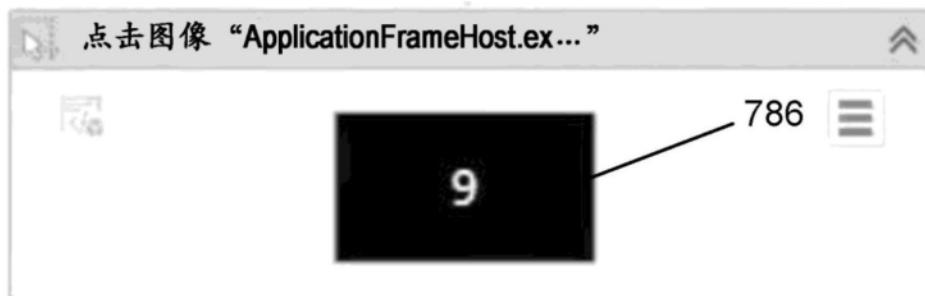


图7Z

760

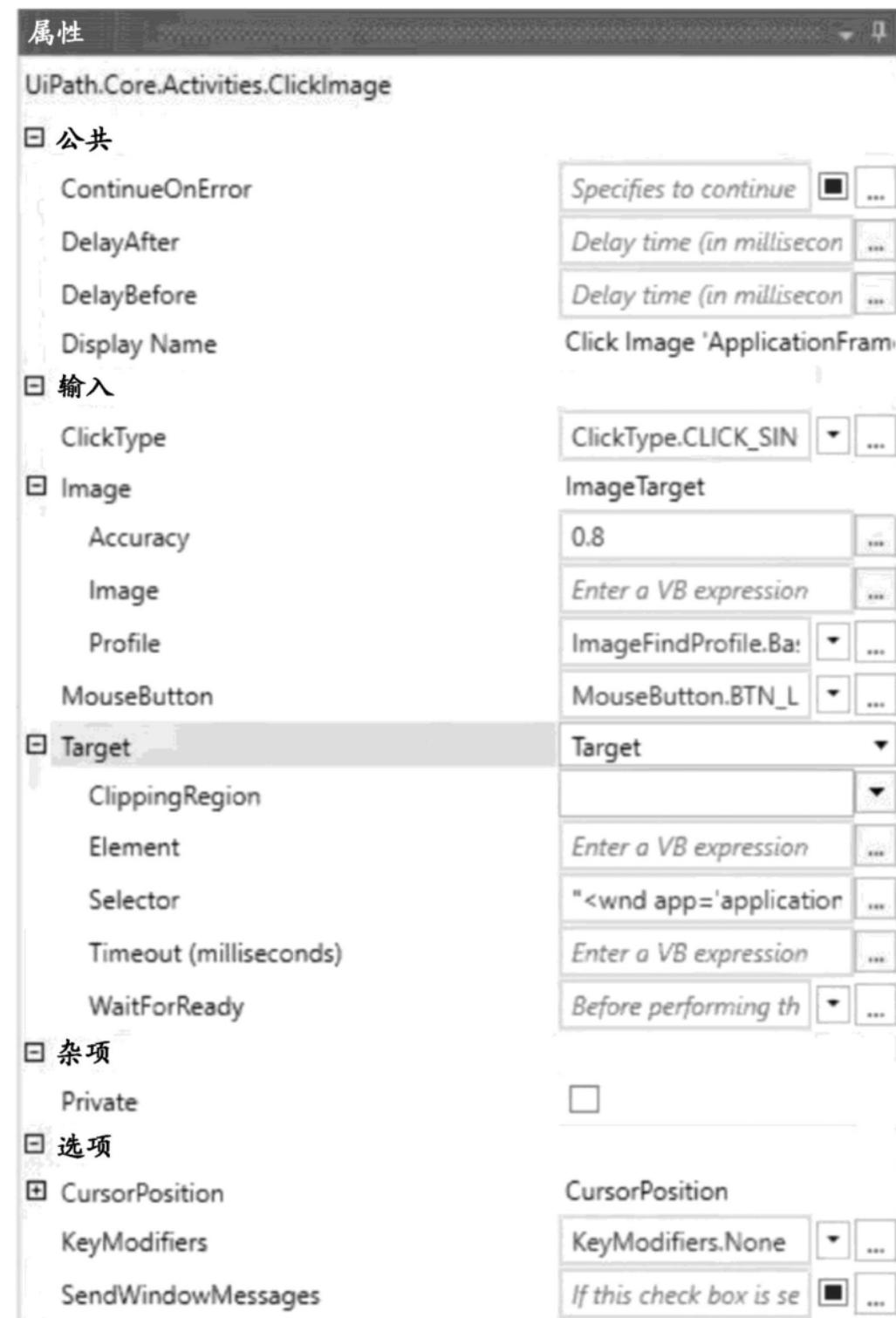


图7a

788

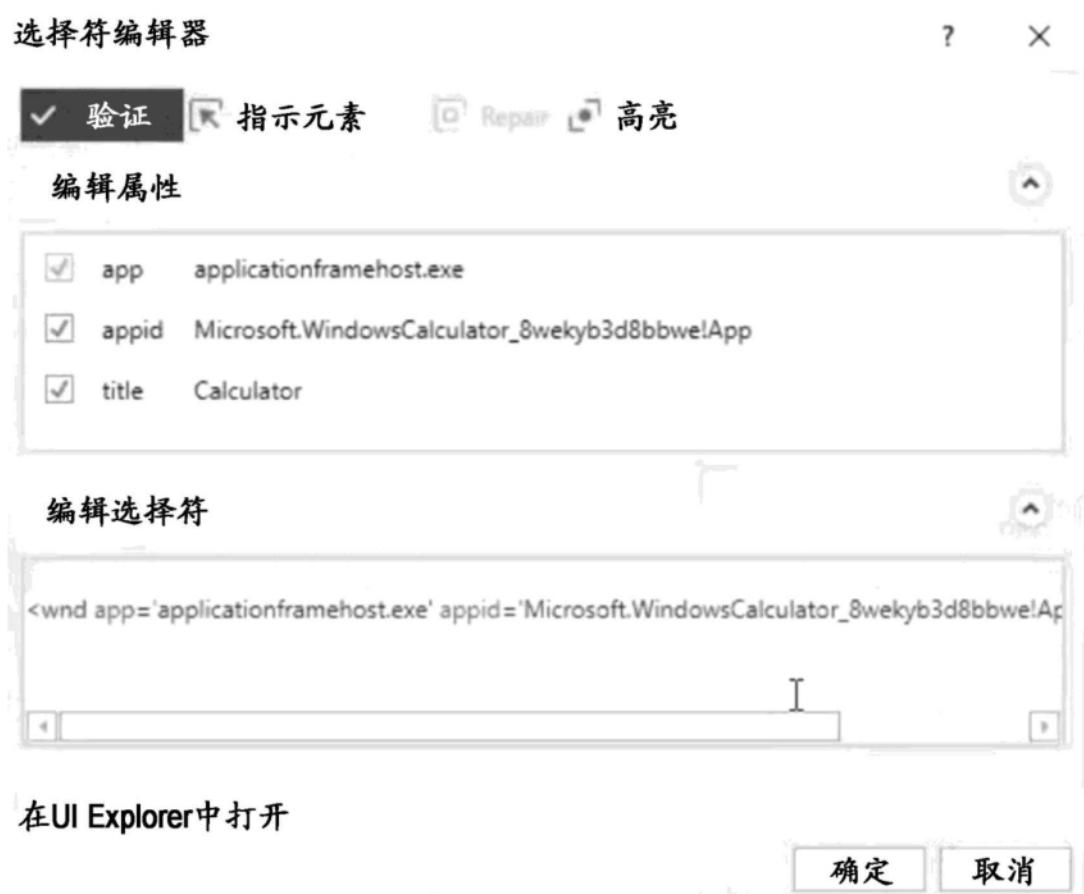


图7b

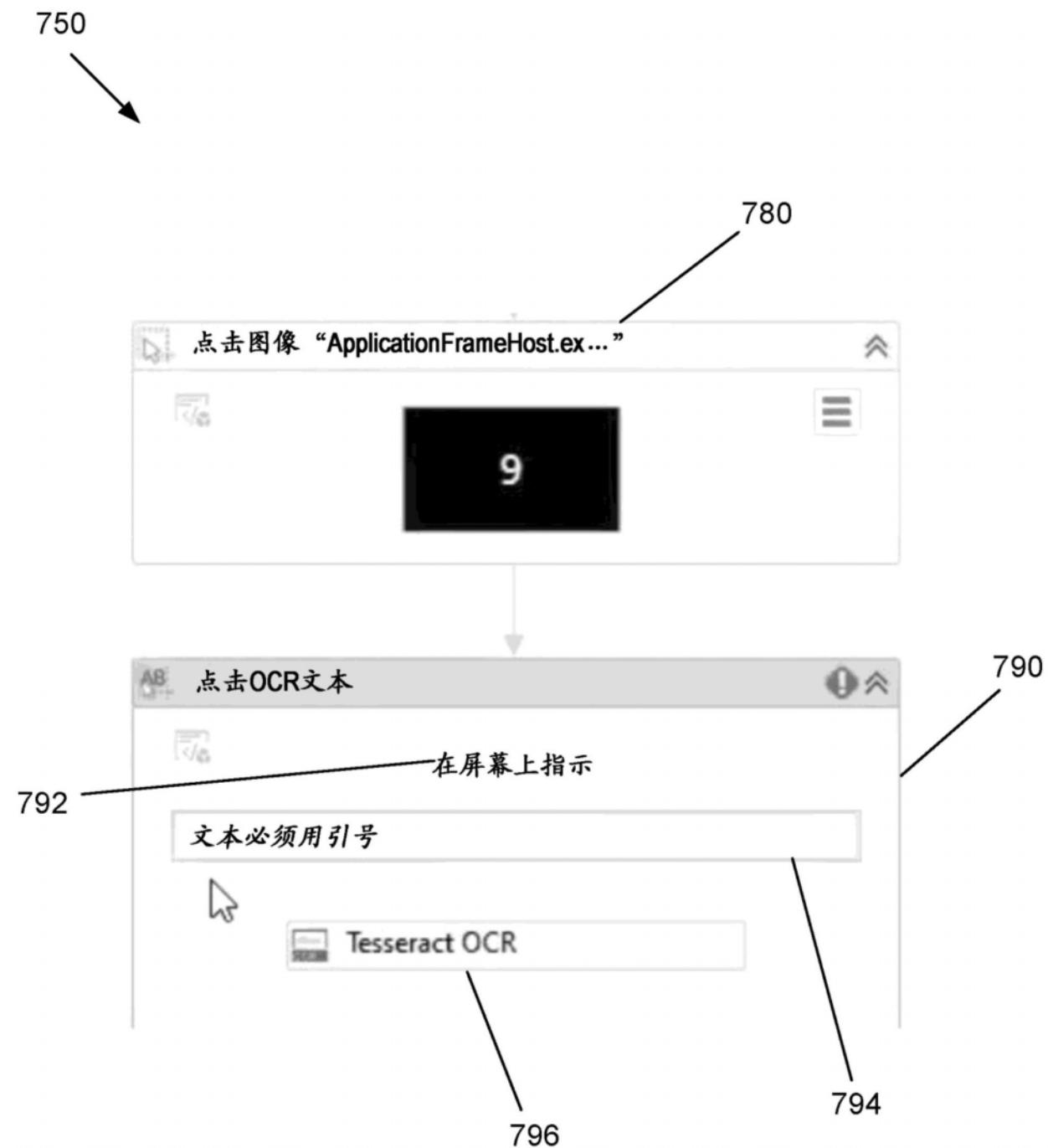


图7c

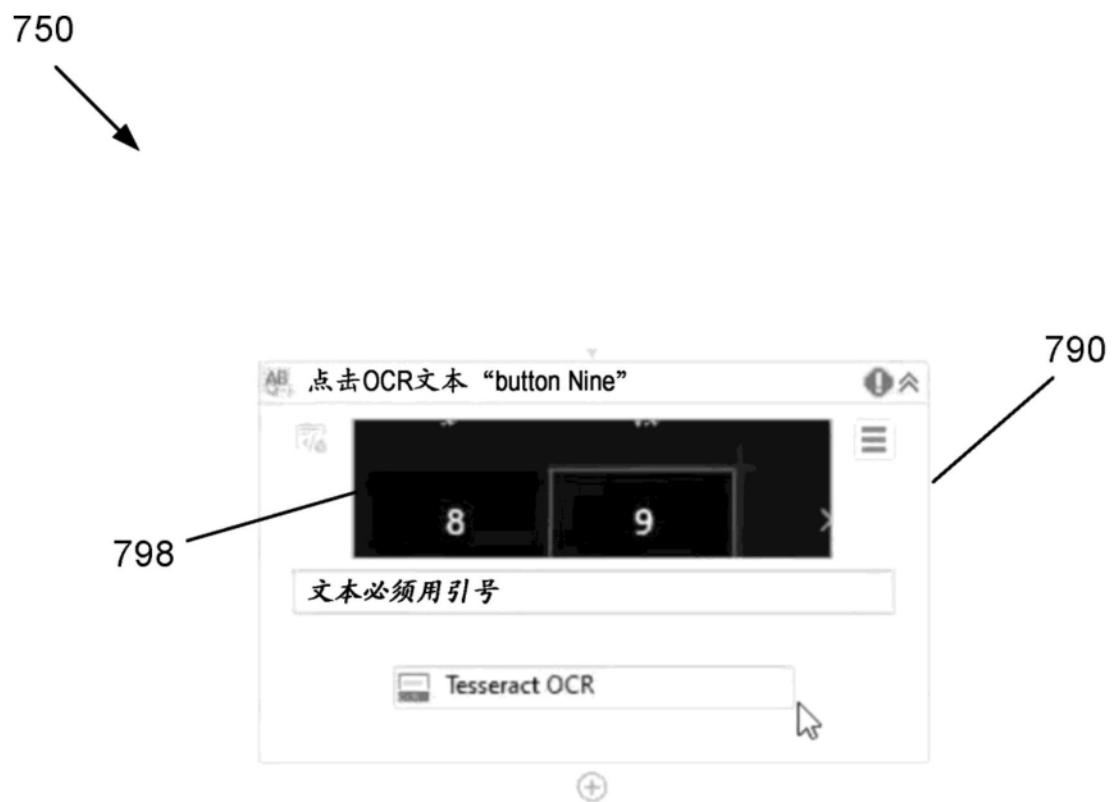


图7d

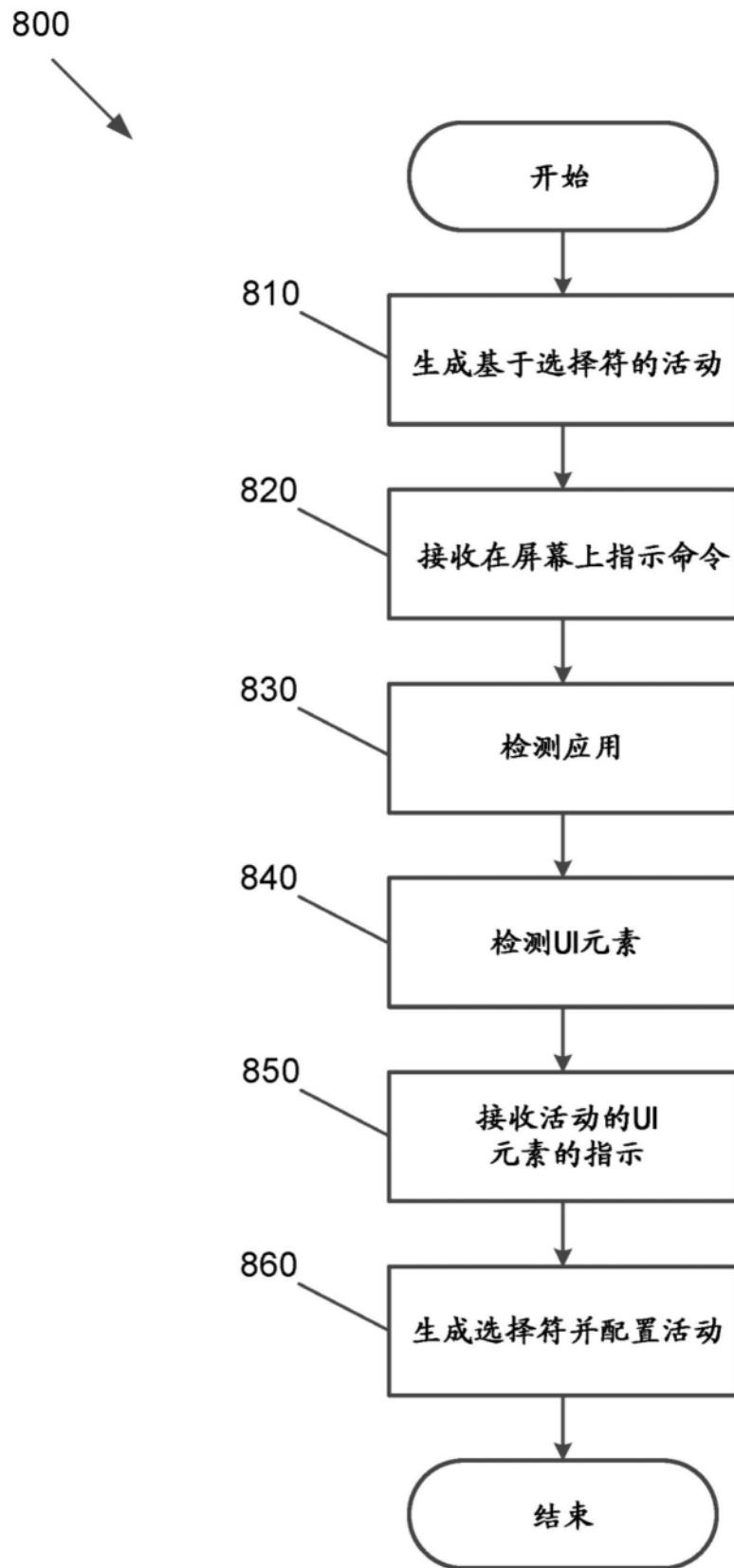


图8

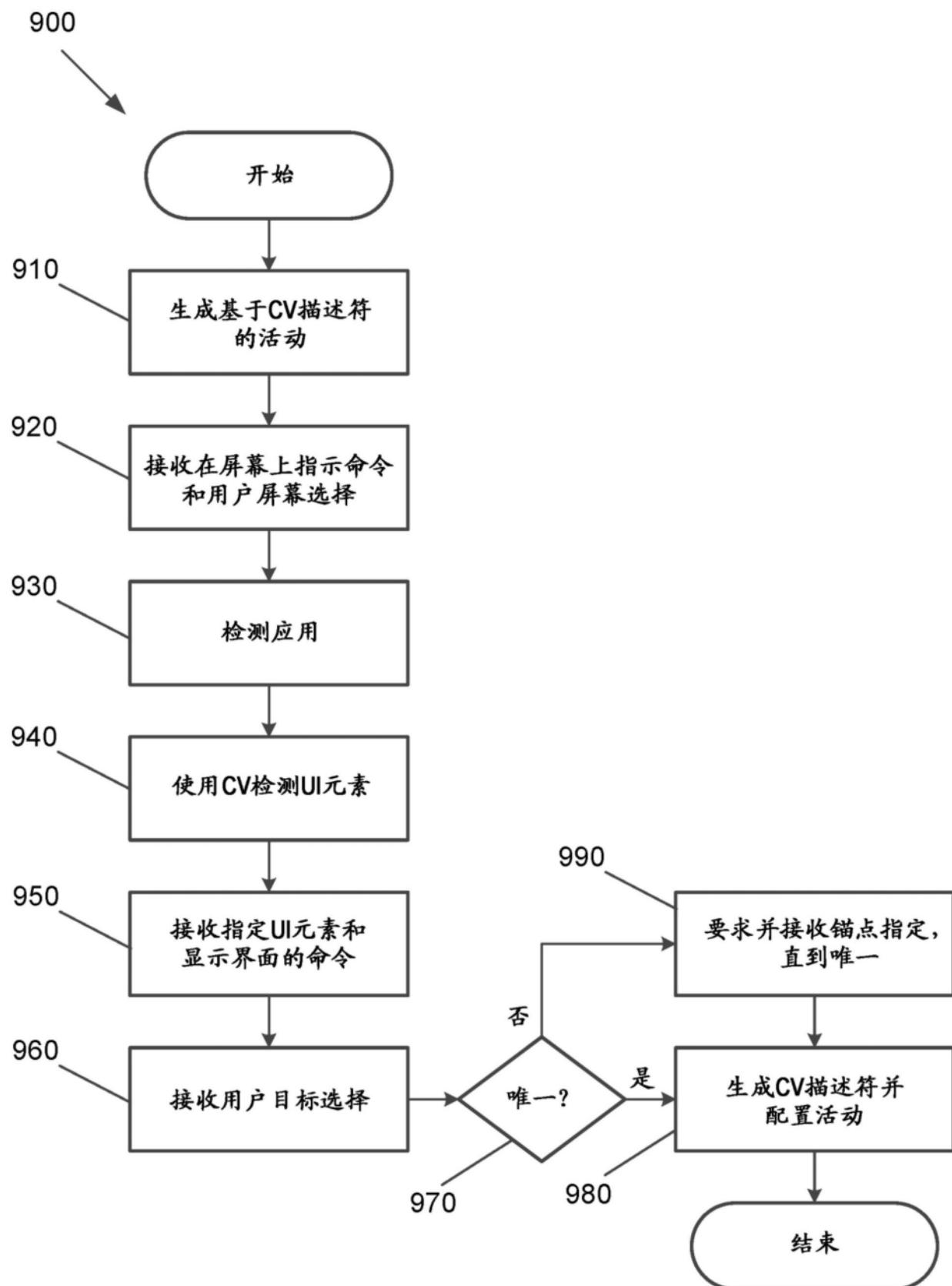


图9

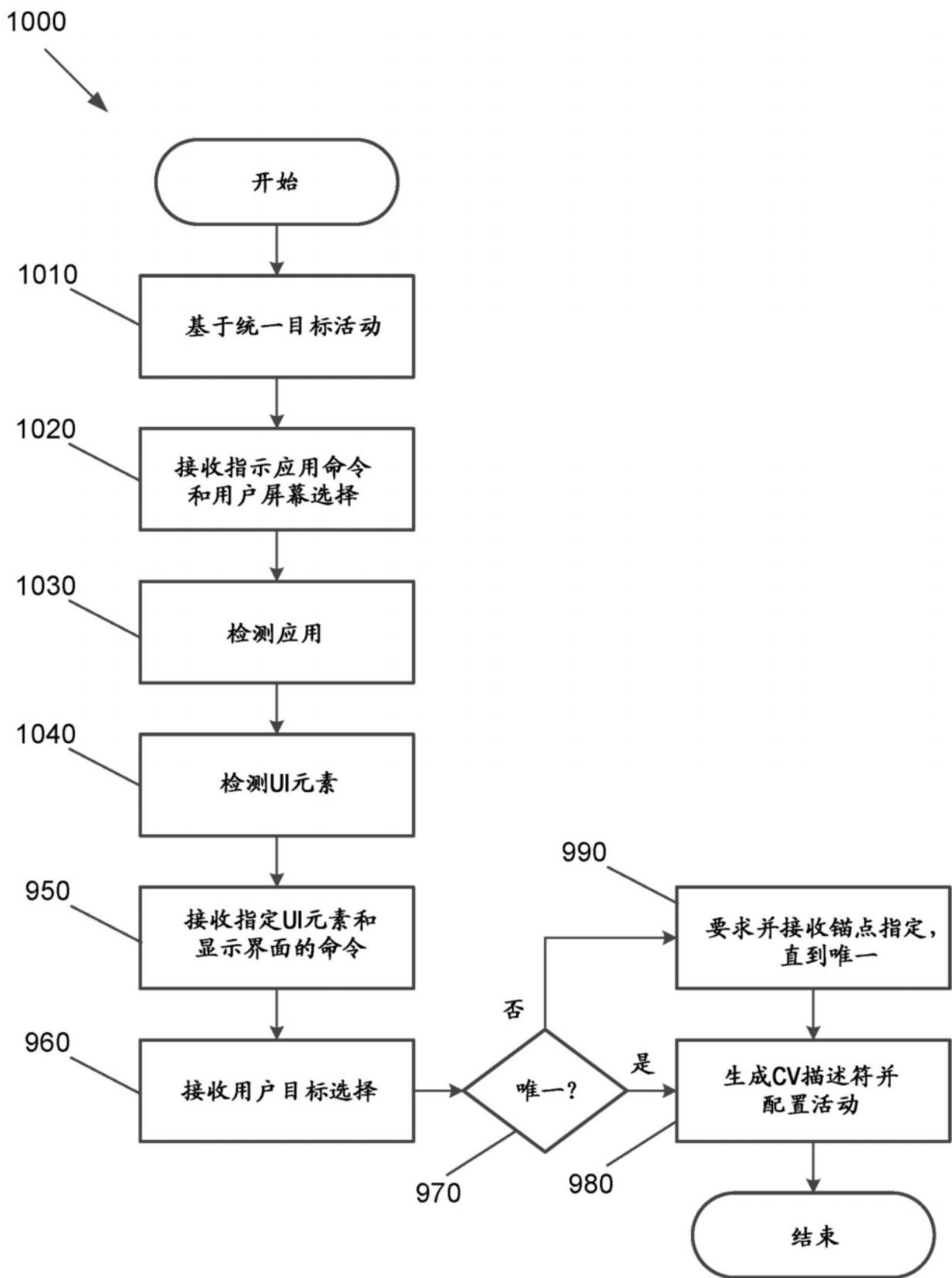


图10

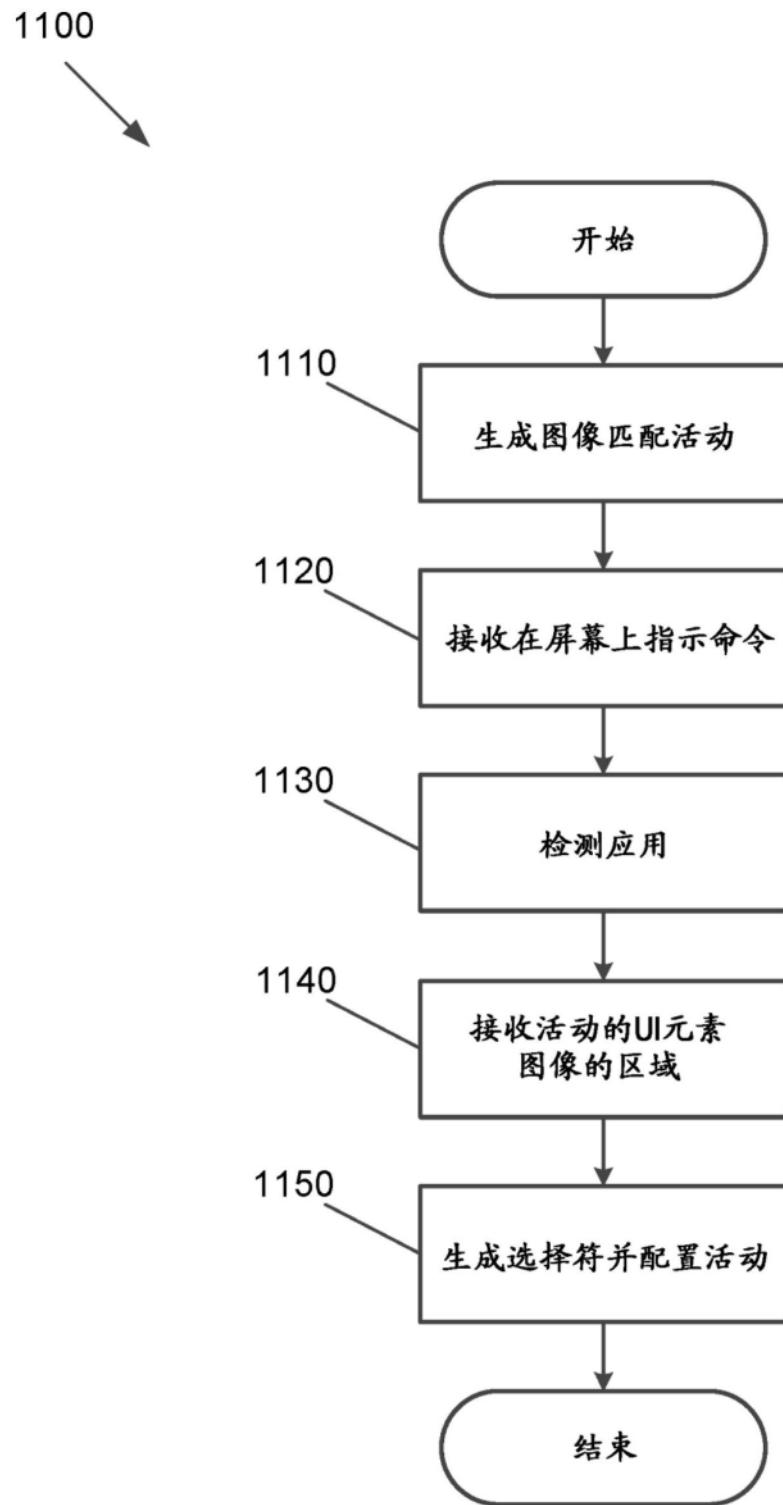


图11

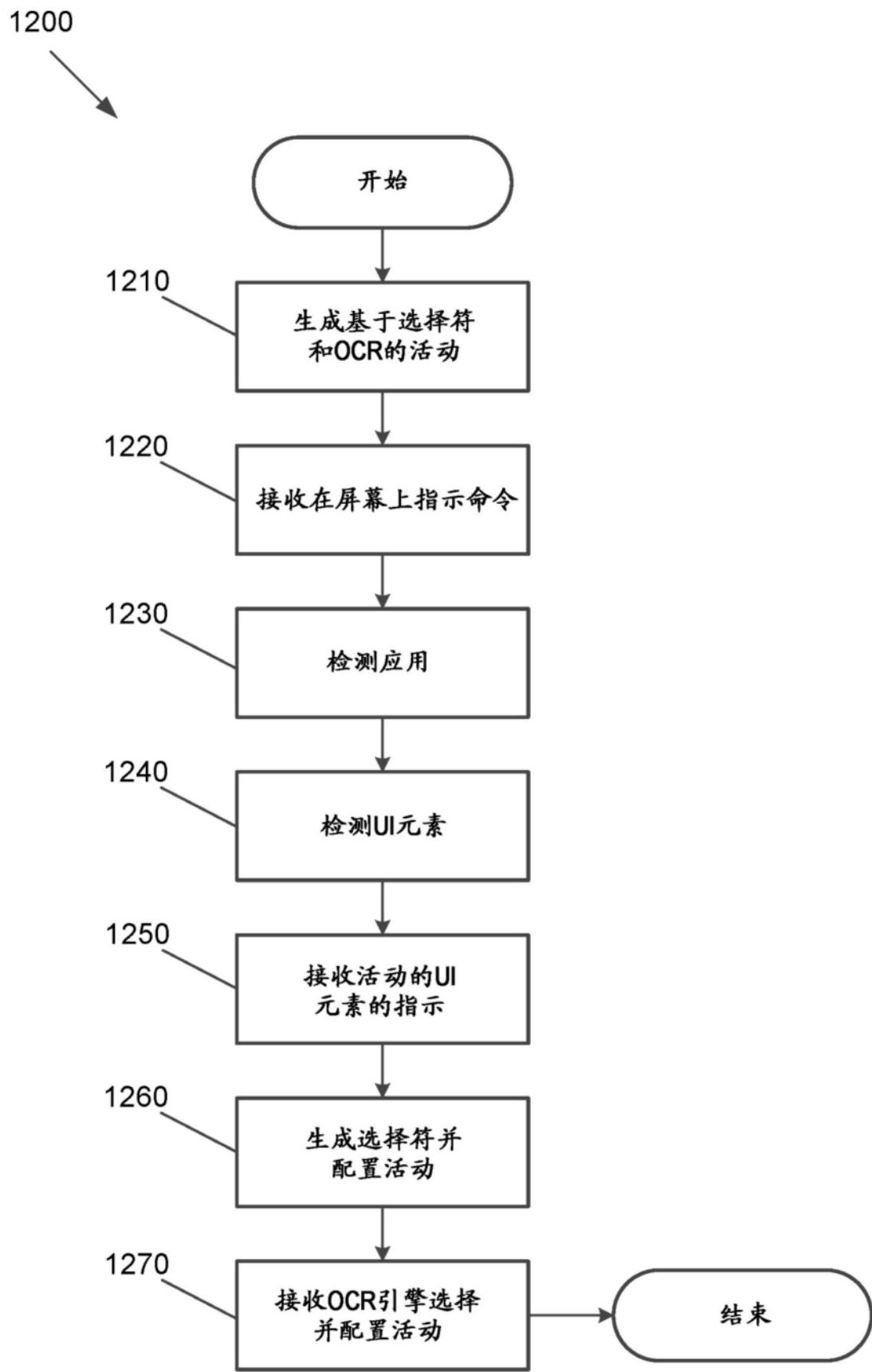


图12

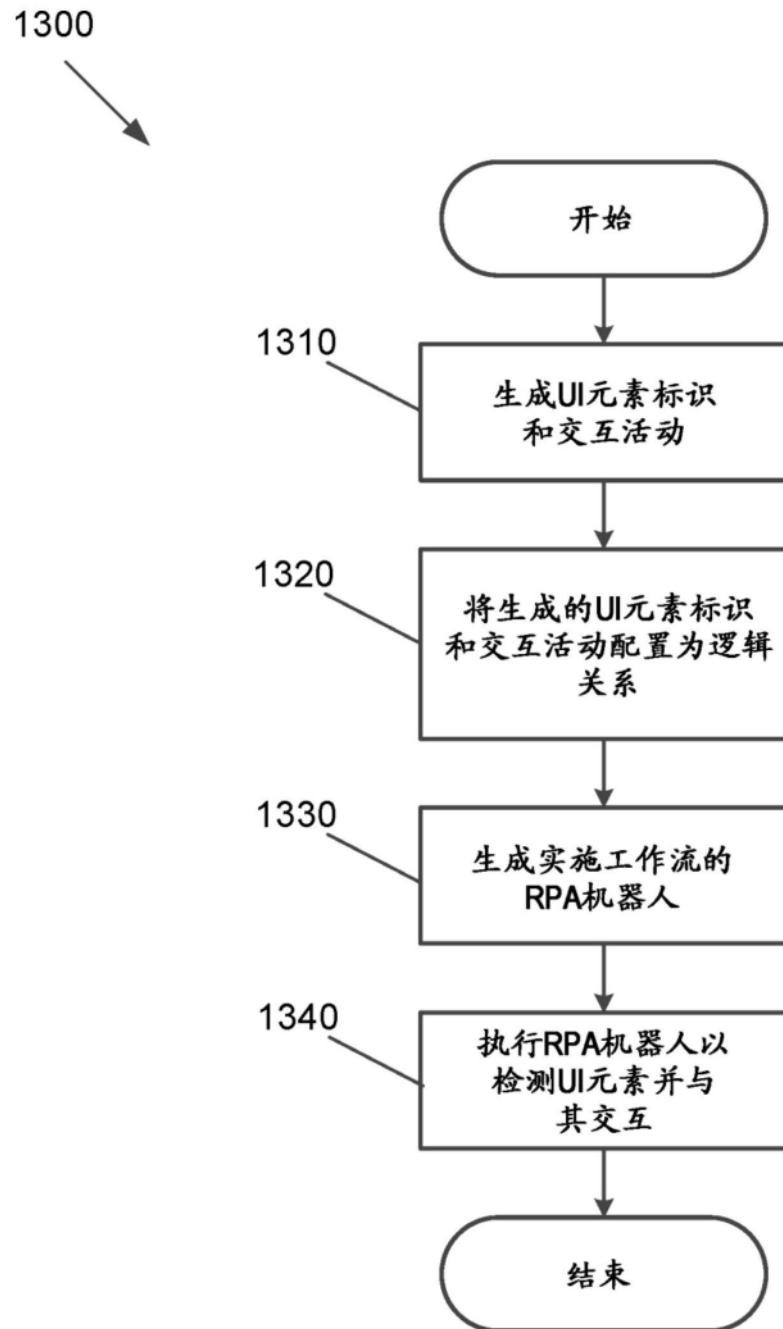


图13

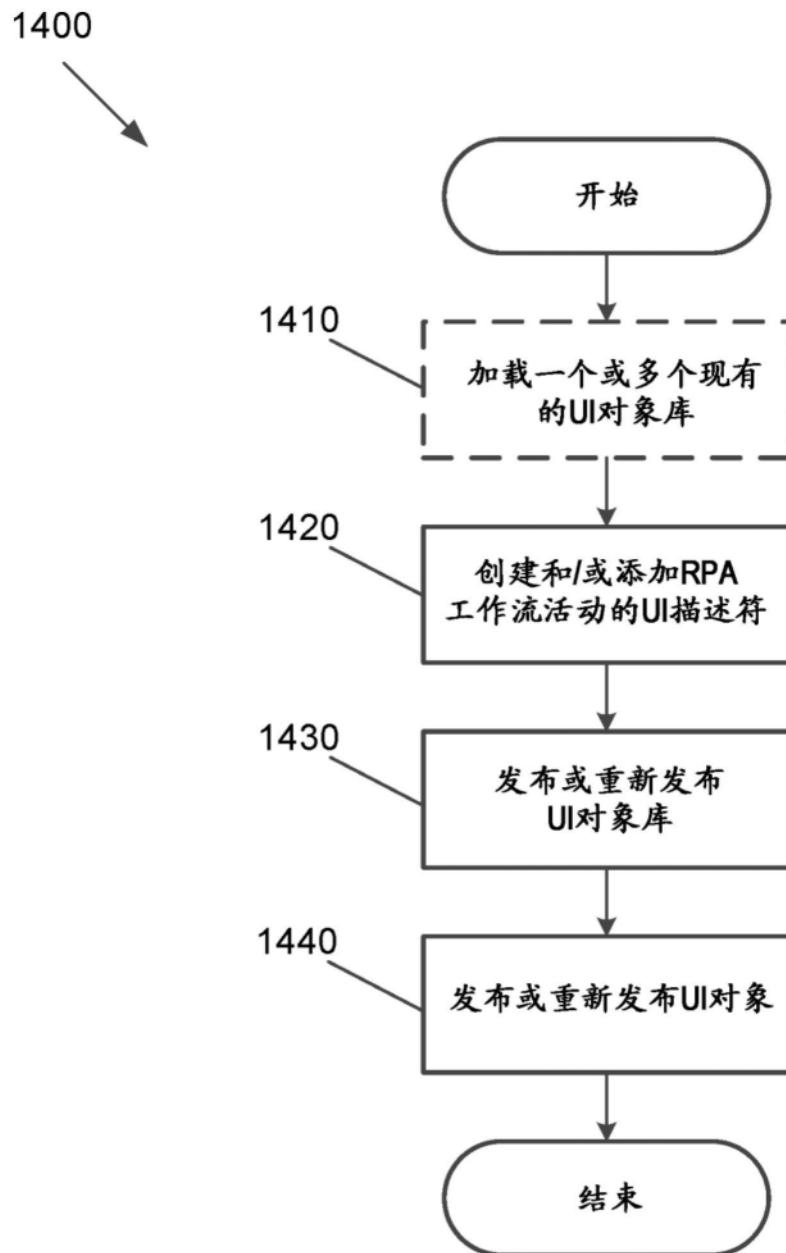


图14