



# (12) 发明专利申请

(10) 申请公布号 CN 113204380 A

(43) 申请公布日 2021.08.03

(21) 申请号 202110494354.6

(22) 申请日 2018.06.29

(30) 优先权数据

2017-128044 2017.06.29 JP

(62) 分案原申请数据

201810695747.1 2018.06.29

(71) 申请人 佳能株式会社

地址 日本东京都大田区下丸子3-30-2

(72) 发明人 丸山直纪

(74) 专利代理机构 北京怡丰知识产权代理有限公司

11293

代理人 李艳丽 齐文文

(51) Int.Cl.

G06F 9/445 (2018.01)

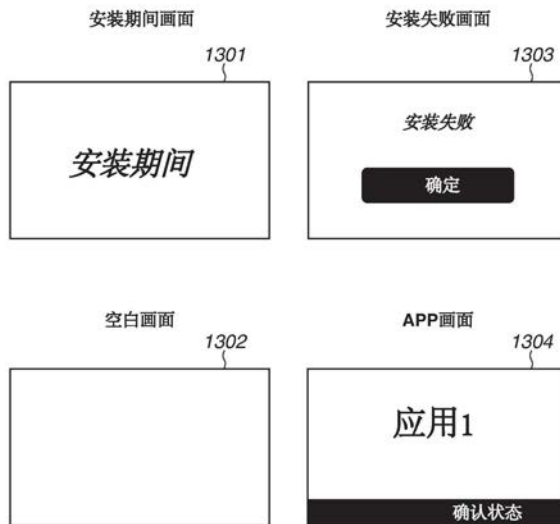
权利要求书2页 说明书12页 附图21页

## (54) 发明名称

信息处理装置和用于信息处理装置的信息处理方法

## (57) 摘要

本发明涉及一种信息处理装置和用于信息处理装置的信息处理方法。所述信息处理装置能与应用服务器、网络服务器和打印机设备进行通信,所述信息处理装置包括:网络浏览器,其被配置为访问所述网络服务器并从所述网络服务器接收包括脚本的网页;其中,所述网络浏览器利用在接收到的网页上输入的打印机设备的访问信息来执行脚本,其中,根据所述网络浏览器对脚本的执行,所述信息处理装置执行的操作包括:从所述打印机设备接收打印机设备信息;将接收到的打印机设备信息发送给所述应用服务器;从所述应用服务器接收至少一个应用程序;以及将接收到的所述至少一个应用程序发送给所述打印机设备。



1. 一种信息处理装置,所述信息处理装置能与应用服务器、网络服务器和打印机设备进行通信,所述信息处理装置包括:

网络浏览器,其被配置为访问所述网络服务器并从所述网络服务器接收包括脚本的网页;

其中,所述网络浏览器利用在接收到的网页上输入的打印机设备的访问信息来执行脚本,

其中,根据所述网络浏览器对脚本的执行,所述信息处理装置执行的操作包括:

从所述打印机设备接收打印机设备信息;

将接收到的打印机设备信息发送给所述应用服务器;

从所述应用服务器接收至少一个应用程序;以及

将接收到的所述至少一个应用程序发送给所述打印机设备。

2. 根据权利要求1所述的信息处理装置,其中,基于在所述网页上接收的用于连接到所述打印机设备的连接指令来执行脚本。

3. 根据权利要求1所述的信息处理装置,其中,所述应用服务器和所述网络服务器是相同的服务器。

4. 根据权利要求1所述的信息处理装置,所述信息处理装置还包括:

显示器;

其中,在所述显示器上显示与对应于打印机设备信息的至少一个应用程序有关的信息。

5. 根据权利要求4所述的信息处理装置,其中,从所述应用服务器接收经由所述显示器指定的应用程序。

6. 根据权利要求1所述的信息处理装置,其中,所述网页包括用于输入所述打印机设备的互联网协议地址的表单。

7. 根据权利要求1所述的信息处理装置,其中,在所述网页上显示安装在所述打印机设备中的应用程序,并且接收用于启动该应用程序的指令。

8. 根据权利要求1所述的信息处理装置,其中,在所述网页上显示安装在所述打印机设备中的应用程序,并且接收用于卸载该应用程序的指令。

9. 根据权利要求1所述的信息处理装置,其中,所述网页包括用于连接到所述打印机设备的按钮。

10. 一种用于信息处理装置的信息处理方法,所述信息处理装置能与应用服务器、网络服务器和打印机设备进行通信,所述信息处理装置具有网络浏览器,

其中,所述网络浏览器访问所述网络服务器并从所述网络服务器接收包括脚本的网页;

所述网络浏览器利用在接收到的网页上输入的打印机设备的访问信息来执行脚本,

其中,根据所述网络浏览器对脚本的执行,所述信息处理方法包括:

从所述打印机设备接收打印机设备信息;

将接收到的打印机设备信息发送给所述应用服务器;

从所述应用服务器接收至少一个应用程序;以及

将接收到的所述至少一个应用程序发送给所述打印机设备。

11. 根据权利要求10所述的信息处理方法,其中,基于在所述网页上接收的用于连接到所述打印机设备的连接指令来执行脚本。

12. 根据权利要求10所述的信息处理方法,其中,所述应用服务器和所述网络服务器是相同的服务器。

13. 根据权利要求10所述的信息处理方法,所述信息处理装置还具有显示器;  
其中,在所述显示器上显示与对应于打印机设备信息的至少一个应用程序有关的信息。

14. 根据权利要求13所述的信息处理方法,其中,从所述应用服务器接收经由所述显示器指定的应用程序。

15. 根据权利要求10所述的信息处理方法,其中,所述网页包括用于输入所述打印机设备的互联网协议地址的表单。

16. 根据权利要求10所述的信息处理方法,其中,在所述网页上显示安装在所述打印机设备中的应用程序,并且接收用于启动该应用程序的指令。

17. 根据权利要求10所述的信息处理方法,其中,在所述网页上显示安装在所述打印机设备中的应用程序,并且接收用于卸载该应用程序的指令。

18. 根据权利要求10所述的信息处理方法,其中,所述网页包括用于连接到所述打印机设备的按钮。

## 信息处理装置和用于信息处理装置的信息处理方法

[0001] 本申请是申请日为2018年06月29日、申请号为201810695747.1、发明名称为“服务器装置、信息处理装置和用于安装应用的控制方法”的发明专利申请的分案申请。

### 技术领域

[0002] 本发明涉及一种信息处理装置和用于信息处理装置的信息处理方法,所述信息处理装置用于与能够提供应用程序的外围设备通信。

### 背景技术

[0003] 存在一种信息处理装置,该信息处理装置能够与信息处理装置上运行的嵌入程序分开地安装单独的可加载程序(app),并使各种服务运行。作为安装app的技术,这样的信息处理装置从信息处理装置外部获取app并安装所述app(日本专利特开第2002-287990号公报)。

[0004] 日本专利特开第2002-287990号公报讨论了一种使用网络浏览器在信息处理装置上安装app并使app有效的技术。

[0005] 传统技术需要用户进行许多操作并且复杂,因此用户要花费时间来安装app。例如,用户需要预先下载app并且写入app名称以及个人计算机(PC)中保存的app包的文件路径。此外,app的安装和有效化不是单组处理,因此用户需要花费更多时间来使app有效。因此,传统技术占用用户的时间安装并使app有效。这可能是用户安装app的障碍。

### 发明内容

[0006] 根据本发明的一个方面,提供一种信息处理装置,所述信息处理装置能与应用服务器、网络服务器和打印机设备进行通信,所述信息处理装置包括:网络浏览器,其被配置为访问所述网络服务器并从所述网络服务器接收包括脚本的网页;其中,所述网络浏览器利用在接收到的网页上输入的打印机设备的访问信息来执行脚本,其中,根据所述网络浏览器对脚本的执行,所述信息处理装置执行的操作包括:从所述打印机设备接收打印机设备信息;将接收到的打印机设备信息发送给所述应用服务器;从所述应用服务器接收至少一个应用程序;以及将接收到的所述至少一个应用程序发送给所述打印机设备。

[0007] 根据以下参照附图对示例性实施例的描述,本发明的其他特征将变得清楚。

### 附图说明

[0008] 图1是例示根据本发明的示例性实施例的系统的示例性配置图。

[0009] 图2是例示根据本发明的示例性实施例的多功能外围设备的硬件的示例性配置图。

[0010] 图3是例示根据本发明的示例性实施例的只读存储器(ROM)的数据区域的示例性配置。

[0011] 图4是例示根据本发明的示例性实施例的应用(app)提供网页的示例性配置图。

- [0012] 图5是例示根据本发明的示例性实施例的app服务器的示例性配置图。
- [0013] 图6是例示根据本发明的示例性实施例的app包的示例性配置图。
- [0014] 图7是例示根据本发明的示例性实施例的app设置文件的示例图。
- [0015] 图8是例示根据本发明的示例性实施例的设备类型信息的示例图。
- [0016] 图9是例示在根据本发明的示例性实施例安装app时自动启动所述app的条件的图。
- [0017] 图10是例示根据本发明的示例性实施例的app提供页面首页画面的示例图。
- [0018] 图11是例示在不存在已安装app的情况下的app提供页的app安装管理画面的示例图。
- [0019] 图12是例示在存在已安装app的情况下的app提供页的app安装管理画面的示例图。
- [0020] 图13是例示在app安装期间要显示的画面的示例图。
- [0021] 图14是例示浏览器获取设备类型信息和已安装app包列表的流程图。
- [0022] 图15是例示关于图14中的步骤S1406中的app包列表获取处理的细节图。
- [0023] 图16是例示浏览器从app服务器下载用户选择的app包并将所述app包传输到外围设备的流程图。
- [0024] 图17是例示在完成app安装之后要通过浏览器执行的流程图。
- [0025] 图18是例示app服务器生成可安装的app包的可安装app包列表的流程图。
- [0026] 图19是例示app服务器将app包传输到浏览器的流程图。
- [0027] 图20是例示将根据本发明的示例性实施例的设备类型信息传输到浏览器的流程图。
- [0028] 图21是例示接收根据本发明的示例性实施例的app包,并执行app包的安装的流程图。
- [0029] 图22是例示根据本发明的示例性实施例app包的安装流程图。
- [0030] 图23是例示根据本发明的示例性实施例的app设置处理的流程图。
- [0031] 图24是例示根据本发明的示例性实施例的app自动启动确定的流程图。
- [0032] 图25是根据本发明的示例性实施例的客户端个人计算机(PC)、app服务器装置和网络服务器装置中每一者的硬件配置的示例图。

### 具体实施方式

- [0033] 下文将参照附图描述用于实施本发明的示例性实施例的示例。
- [0034] 以外围设备是多功能外围设备为例。图像形成装置是外围设备的示例。外围设备的示例包括智能手机和具有音频响应功能或人工智能(AI)功能的嵌入设备。图像形成装置可以是电子照相多功能外围设备、单一功能打印机或喷墨打印机。图1是例示根据本示例性实施例的系统的示例性配置的框图。多功能外围设备(100)和浏览器(101)可经由网络2003彼此通信。浏览器(101)和应用(app)提供网页(102)可经由网络2003彼此通信。浏览器(101)和app服务器(103)可经由网络2003彼此通信。app服务器(103)包括在app服务器装置2001中。浏览器(101)包括在客户端个人计算机(PC)2000中。app提供网页(102)包括在网络服务器装置2002中。如图1中所示,各装置中的服务经由网络2003通信。此时,app服务器

(103) 和app提供网页 (102) 可经由浏览器 (101) 与多功能外围设备 (100) 通信。具体而言,浏览器 (101) 获取app提供网页 (102) 并使用中央处理单元 (CPU) 执行app提供网页 (102) 中保存的程序或链接目的地处的程序。以此方式,app服务器 (103) 和多功能外围设备 (100) 可经由浏览器 (101) 彼此进行双向通信。可选地,app服务器装置2001可以与网络服务器装置2002集成在一起。在该情况下,应理解,装置中的多个程序将内部获取、传输及接收各种信息。

[0035] 图2是例示多功能外围设备 (100) 的示例性配置的图。在多功能外围设备 (100) 中,CPU (201) 是用于控制系统的控制单元,并控制整个多功能外围设备 (100)。只读存储器 (ROM) (202) 包括诸如嵌入式多媒体卡 (EMMC) 的闪存存储器以允许读写操作,并且存储CPU (201) 的控制程序。ROM (202) 还存储设置值和图像数据,以便设置值和图像数据在多功能外围设备 (100) 关闭时不会被擦除。ROM (202) 还存储app包数据。动态随机存储存储器 (DRAM) (203) 存储程序控制变量并且可以存储由CPU (201) 执行的指令或程序。DRAM (203) 是能够临时保存要处理的图像数据的易失性存储器。操作单元 (204) 是用于显示设备中的信息的、与用户的接口单元。通信单元 (205) 是设备与外部通信网络之间的接口单元。通信单元 (205) 包括网络通信单元,所述网络通信单元是与网络2003的接口。扫描仪 (206) 是用于读取图像数据并将所述图像数据转换成二进制数据的设备。使用扫描仪 (206) 来进行图像传输功能的原稿读取。打印机 (207) 是用于将图像数据输出到记录片材的设备。这些控制单元经由数据总线 (208) 连接在一起。

[0036] 使用以下示例进行描述:此多功能外围设备 (100) 从多功能外围设备 (100) 外部与在多功能外围设备 (100) 的主机上运行的主要程序分开地获取作为单独的可加载程序的app,将所述app安装在多功能外围设备 (100) 上而无需用户干预,并将所述app变更为有效状态,有效状态是能够执行app的状态。

[0037] 图25是例示客户端PC 2000、app服务器装置2001和网络服务器装置2002中的每一者的硬件配置的示例的图。这些装置中的每一者均包括网络接口卡 (NIC) 2502、CPU 2503和ROM 2504。此外,这些装置中的每一者均包括随机存储存储器 (RAM) 2505、硬盘驱动器 (HDD) 2506和输入/输出 (I/O) 2507。NIC 2502是用于与多功能外围设备 (100) 通信的设备。NIC 2502连接至图1中的网络2003。可以使用调制解调器代替NIC 2502而使装置连接至公用线路。ROM 2504存储控制程序。RAM2505用作工作存储器并且可以存储指令或从ROM 2504传输的控制程序。HDD 2506存储本示例性实施例的附图中所示的各种数据结构。I/O 2507将数据传输到装置外部及从装置外部接收数据。CPU 2503通过执行存储在诸如ROM2504和RAM2505中的指令来控制NIC 2502、ROM 2504、RAM 2505、HDD 2506及I/O 2507进行以下描述的操作。显示器 (DSP) 2508是诸如液晶显示器或阴极射线管 (CRT) 显示器的显示设备,并且显示各种处理结果。CPU 2503控制根据本示例性实施例的要通过客户端PC 2000、app服务器装置2001和网络服务器装置2002执行的处理。

[0038] 图3是例示多功能外围设备 (100) 中的ROM (202) 的数据区域的示例性配置图。根据用途将ROM (202) 分成多个区域。在ROM (202) 中作为所述多个区域,定义例如用于使多功能外围设备 (100) 的主机运行的主要程序区域 (301) 以及用于保存app并使所述app运行的app程序区域 (302)。

[0039] 图4是例示为用户提供app的app提供网页 (102) 的示例性配置。App提供网页 (102)

的信息存储在app服务器2001的HDD 2506中。App提供网页(102)包括提供安装app所需的处理(诸如,下载和传输app)的app安装脚本(401)以及在页面上显示的超文本标记语言(HTML)文件(402)。如果浏览器(101)访问app提供网页(102),则app提供网页(102)经由网络2003为浏览器(101)提供app安装脚本(401)。脚本(401)可通过JavaScript(注册商标)实现。通过浏览器(101)执行图4中的脚本(401)。然后,脚本(401)可配置为执行图14中步骤S1404及此后的处理以及图15、图16和图17中的处理。脚本(401)可以从录入窗体(将在图10中描述)获得它的输入参数并生成设备类型信息和安装app包列表。

[0040] 图5是例示app服务器(103)的示例性配置图。app服务器(103)包括为多功能外围设备(100)提供的多个app包(501)和设备类型/app包对应列表(502)。如上所述,app服务器(103)存储在app服务器装置2001的HDD 2506中。

[0041] 图6是例示参照图5所述的各app包(501)和设备类型/app包对应列表(502)的示例性配置图。app包(501)存储在app服务器装置2001的HDD 2506中。app包(501)的示例性配置(600)包括通过压缩要通过多功能外围设备(100)执行的一组app程序获得的app归档(601)以及验证app包(501)的有效性的签名(605)。app归档(601)包括控制app操作的app程序(602)、要在app中显示图像的app资源文件(603)以及写入app的操作设置的app设置文件(604)。设备类型/app包对应列表(502)的示例(606)具有包括设备类型名称(801)和能够在与设备类型名称(801)对应的设备类型上安装的app包的名称的项目。

[0042] 图7是例示参照图6描述的app设置文件(604)的内容示例图。在图7中,以三个设置项作为示例。首先,第一设置项为用于设置是否能够卸载app的“uninstallableFlag”(701)。其次,第二设置项为用于设置app的冻结决定时间的“runawayDecisionSec”(702)。最后,第三设置项为用于设置是否在安装app时自动使app有效的“defaultEnableFlag”(703)。当使“defaultEnableFlag”(703)有效时,在安装app之后,自动使app在多功能外围设备(100)中有效。

[0043] 图8是例示多功能外围设备(100)的设备类型信息(800)的示例图。在该情况下,图8以多功能外围设备(100)的设备类型名称(801)和表示例如多功能外围设备(100)是否为收费设备或者是否可使用扫描仪(206)的型号信息(802)为例进行例示。图8中的设备类型信息(800)存储在多功能外围设备(100)的ROM 202中。型号信息(802)包括表示多功能外围设备(100)是否为应针对在该型号中提供的应用收费的型号的信息。虽然在设备类型名称(801)中写入了“STERA\_XN”,但也可以写入任意识别信息。例如,在“STERA\_XN”的“N”中,可存储等于或大于1的整数,诸如“STERA\_X1”或者“STERA\_X2”。同样,在型号信息(802)的“X”中,可以将识别任意多功能外围设备型号的信息存储为“MODEL\_X”或“MODEL\_Y”。型号信息(802)还包括表示多功能外围设备(100)是否为针对例如应用收费的收费型号的信息。

[0044] 图9是例示在安装app时自动启动所述app的条件(900)图。如果在这些条件中自动启动为“是”,则安装后将自动启动或自动有效化app。例如,将“defaultEnableFlag”(703)设置为“假”或“真”。在将“defaultEnableFlag”(703)设置为“假”的情况下,不管当前安装是否为新安装,都不设置最近更新后的应用的自动启动(“否”)。在将“defaultEnableFlag”(703)为“真”的情况下,如果当前安装不是新安装(存在应更新的现有已安装程序),除非应用状态为“无效”,否则将设置自动启动。以下将参照图24详细描述该处理的具体示例。虽然在图9中进行了定义,但也可以不同地定义条件(900)。然后,可以与图24不同地进行操作。

在如下两种情况下自动启动为“是”：(1) “defaultEnableFlag” (703) 设置为“真”，当前安装不是新安装并且应用状态为“有效”；以及(2) “defaultEnableFlag” (703) 设置为“真”，当前安装为新安装并且应用状态为“无效”或者“有效”。

[0045] 图10是例示app提供页面首页画面(1000)的图。在此画面上，提供用于输入关于连接到网络2003的多功能外围设备(100)的三条信息的窗体以及单个按钮。首先，第一个窗体为用于输入多功能外围设备(100)的互联网协议(IP)地址或域名系统(DNS)名称的窗体(1001)。其次，第二个窗体为用于输入多功能外围设备(100)的管理人员识别(ID)的窗体(1002)。最后，第三个窗体为用于输入多功能外围设备(100)的管理人员密码的窗体(1003)。然后，app提供页面首页画面(1000)包括用于通过浏览器(101)将这些信息从app提供网页(102)传输到多功能外围设备(100)的设备连接按钮(1004)。当将app提供网页(102)的统一资源定位符(URL)输入到PC(2000)中的浏览器(101)的URL输入部时，app提供页面首页画面(1000)进行以下处理。即，浏览器(101)访问网络服务器装置2002中的app提供网页(102)并且获得图4中所示的数据。然后，在PC 2000的显示设备上显示图10中的画面。在图10中的画面上，如果将输入提供到所有输入部1001、1002和1003，并且按下设备连接按钮(1004)，则执行图4中的脚本(401)。作为结果，显示图11或图12中的画面。下文将详细描述该处理。

[0046] 可选地，可以进行以下处理。首先，客户端PC 2000的浏览器(101)可以访问嵌入在多功能外围设备(100)中的网络服务。在浏览器(101)正在访问网络服务的同时，浏览器(101)可以显示用于管理多功能外围设备(100)的管理画面(未示出)。如果在管理画面的预定画面上点击链接，则预定画面可以跳转到app安装管理画面1100。

[0047] 图11是例示在多功能外围设备(100)中不存在已安装app的情况下，在app提供网页(102)上显示的app安装管理画面(1100)图。该画面显示app服务器(103)所拥有的app包(501)中可安装在多功能外围设备(100)上的app包的可安装app包列表(1101)。App包列表(1101)具有包括app名称(1102)、app版本(1103)、app包类型(1104)和app包的动作(1105)的项目。用户按下针对app包的动作(1105)的按钮且借此能够安装app。

[0048] 图12是例示在多功能外围设备(100)中存在已安装app的情况下，在app提供网页(102)上显示的app安装管理画面(1200)图。

[0049] 按下图11中针对app包的动作(1105)的按钮，由此执行预定处理，并且图11中的画面转变为图12中的画面。下文将详细描述所述预定处理。图12例示可以卸载(1208)应用1的状态，即在多功能外围设备(100)上安装应用的状态。应用2类似于图11中的应用2。

[0050] 即，在图11中没有显示列表的“已安装包”部中，当前显示多功能外围设备(100)所拥有的已安装app包的已安装app包列表(1201)。已安装app包列表(1201)具有五个项目，所述五个项目包括app名称(1202)、app版本(1203)、app包类型(1204)、app有效化/无效化开关(1205)和对多功能外围设备(100)的动作(1206)。用户操作app有效化/无效化开关(1205)，由此能够启动或关闭app。此外，根据包是否已安装或可用(但尚未安装)动作(1206)具有默认按钮。对于可用的包，默认按钮是“安装”。用户按下针对多功能外围设备(100)上的安装的“安装”按钮(图11中的1106或图12中的1207)，由此能够在多功能外围设备(100)上安装app。对于已经安装的包，默认按钮是“卸载”。如果用户按下图12中的“卸载”按钮1208，则从多功能外围设备(100)卸载已安装的应用。即，图12中的状态返回到图11中的状态。

[0051] 图13是例示在app安装期间要显示的画面(1301至1304)的示例图。安装期间画面(1301)是为了在多功能外围设备(100)上正在安装app时不操作操作单元(204)而要显示的画面的示例。空白画面(1302)是在自动启动已安装app直至app启动为止时,要在操作单元(204)上显示的画面的示例。安装失败画面(1303)是在app安装期间出现问题及安装失败时要在操作单元(204)上显示的画面的示例。app画面(1304)是已安装app启动时要在操作单元(204)上显示的画面的示例。

[0052] 接下来,将参照流程图描述关于app的获取到安装的控制方法。通过满足以下条件实现图14至图24中的流程图所示的操作。条件为浏览器(101)加载通过app提供网页(102)提供的app安装脚本(401)。然后,浏览器(101)与多功能外围设备(100)及app服务器(103)通信。

[0053] 可选地,多个浏览器(101)、多个多功能外围设备(100)和app服务器(103)还可协作执行以下流程图所示的处理。可选地,除了CPU,还可以使用硬件电路来执行一些处理。

[0054] 首先,描述图14中的流程图。图14是例示当浏览器(101)从多功能外围设备(100)获取设备类型信息(800)和已安装app包列表(1201)时进行的控制图。首先,在客户端PC 2000中启动浏览器(101)。然后,将网络服务器装置2002的统一资源标识符(URI)输入到浏览器(101)图10中的地址栏。根据输入,在客户端PC 2000中的CPU 2503的控制下开始执行图14中的处理。

[0055] 在步骤S1401中,浏览器(101)访问app提供页面首页画面(1000)。然后,在步骤S1402中,浏览器(101)从app提供网络服务104中获取app提供网页(102)。app提供网页(102)包括安装脚本(401)和HTML文件402。浏览器(101)分析HTML文件402和脚本(401)。然后,浏览器(101)在客户端PC 2000的DSP 2508上显示图10中的app提供页面首页画面(1000)。

[0056] 然后,在步骤S1403中,浏览器(101)在用户按下app提供页面首页画面(1000)上的设备连接按钮(1004)之前进行等待。如果用户将信息输入到录入窗体1001至1003,并按下设备连接按钮(1004)(在步骤S1403中为“是”),则处理进入步骤S1404。

[0057] 在步骤S1404中,浏览器101获取录入窗体(1001至1003)的设置值,并且处理进入步骤S1405。

[0058] 在步骤S1405中,基于在步骤S1404中获取的设置值,通过浏览器101控制的CPU2503执行安装脚本(401)。根据脚本401的执行,浏览器(101)从多功能外围设备(100)获取设备类型信息(800),并且处理进入步骤S1406。

[0059] 在步骤S1406中,CPU 2503基于在步骤S1404中获取的设置值执行安装脚本(401)。在该处理中,浏览器(101)从多功能外围设备(100)获取已安装app包列表(1201)。

[0060] 接下来,将描述图15中的流程图。图15是例示图14中由客户端PC2000进行的app包列表获取处理(步骤S1406)的细节图。浏览器(101)从app服务器(103)获取可安装app包列表(1101),并且当前画面转换为app安装管理画面(1100或1200)。

[0061] 在步骤S1501中,浏览器(101)将从多功能外围设备(100)获取的设备类型信息(800)传输到app服务器(103),并且处理进入步骤S1502。此后,app服务器(103)进行下文描述的图18中的处理。然后,在步骤S1502中,浏览器(101)从app服务器(103)获取可安装在多功能外围设备(100)上的app包的可安装app包列表(1101),并且处理进入步骤S1503。

[0062] 在步骤S1503中,浏览器(101)使用安装脚本(401)的结果和两个app包列表(即可安装在多功能外围设备(100)上的app包的可安装app包列表(1101)和已安装在多功能外围设备(100)上的app包的已安装app包列表(1201))生成app安装管理画面(1100或1200)。在该情况下,首先,如果多功能外围设备(100)上没有安装应用(例如,在安装多功能外围设备(100)时),从app服务器装置2001接收图11中的画面并显示所述画面。此时,如果多功能外围设备(100)上没有安装app包,则不显示已安装app包列表(1201)。此时,可以显示表示“不存在已安装app”的语句的app安装管理画面(1100)。如果多功能外围设备(100)上安装有app包,则显示表示已安装app包列表(1201)的app安装管理画面(1200)。

[0063] 在步骤S1504中,当前画面转换为在步骤S1503中通过浏览器(101)生成的app安装管理画面(图11中的1100或图12中的1200)。此时,假设在多功能外围设备(100)刚一就位之后还没有在多功能外围设备(100)上安装app的情况来描述app的安装。因此,当前画面转换为app安装管理画面(1100)。

[0064] 接下来,将描述图16中的流程图。图16是例示浏览器(101)从app服务器(103)下载用户选择的app包并且将所述app包传输到多功能外围设备(100)的流程图。即,在客户端PC 2000的CPU 2503的控制下执行图16中的处理。通过用户在步骤S1504中按下针对app包的动作(1105或1207)的按钮开始该流程。

[0065] 在步骤S1601中,浏览器(101)将app包获取请求传输到app服务器(103)以获取作为用户选择的下载目标的app包,并且处理进入步骤S1602。同时,app服务器(103)进行图19中的处理。

[0066] 然后,在步骤S1602中,浏览器(101)从app服务器(103)接收目标app包(501),并且处理进入步骤S1603。

[0067] 在步骤S1603中,浏览器(101)将目标app包(501)传输到多功能外围设备(100)。

[0068] 接下来,将描述图17中的流程图。图17是例示在完成app包(501)安装之后,将由浏览器(101)执行的流程图。即,在客户端PC 2000中的CPU 2503的控制下执行图17中的处理。在从多功能外围设备(100)接收到安装结果通知的情况下开始该流程。在步骤S1701中,浏览器(101)从多功能外围设备(100)获取已安装app包列表(1201),并且处理进入步骤S1702。在步骤S1702中,浏览器(101)使用在步骤S1701中获取的已安装app包列表(1201)更新app安装管理画面(1100或1200)。

[0069] 接下来,将描述图18中的流程图。图18是例示app服务器(103)生成可安装在多功能外围设备(100)上的app包的可安装app包列表(1101)的流程图。即,通过app服务器装置2001中的CPU 2503执行app服务器(103)实现图18中的处理。

[0070] 在步骤S1801中,app服务器(103)从浏览器(101)中接收多功能外围设备(100)的设备类型信息(800),并且处理进入步骤S1802。此处要接收的设备类型信息(800)是在步骤S1501中从多功能外围设备(100)传输的。

[0071] 在步骤S1802中,基于设备类型信息(800)和设备类型/app包对应列表(502),app服务器(103)从app服务器(103)所拥有的app包(501)生成可安装在多功能外围设备(100)上的app包的可安装app包列表(1101)。具体而言,从设备类型/app包对应列表(502)中选择并生成对应于由设备类型信息(800)表示的设备类型名称(801)的app包的列表。

[0072] 在步骤S1803中,app服务器(103)将生成的可安装app包列表(1101)传输到浏览器

(101)。在步骤S1502中通过多功能外围设备(100)接收该信息。

[0073] 接下来,将描述图19中的流程图。图19是例示app服务器(103)将app包(501)传输到浏览器(101)的流程图。即,通过app服务器装置2001中的CPU 2503执行app服务器(103)实现图19中的处理。

[0074] 在步骤S1901中,app服务器(103)从浏览器(101)中接收app包请求,并且处理进入步骤S1902。在图16中的步骤S1601中,从浏览器(101)将该app包请求传输到app服务器装置2001。

[0075] 在步骤S1902中,app服务器(103)从app服务器装置2001所拥有的app包(501)中选择匹配app包请求的app包(501)。然后,app服务器(103)将选择的app包(501)传输到浏览器(101)。

[0076] 接下来,将描述图20中的流程图。图20是例示多功能外围设备(100)将设备类型信息(800)传输到浏览器(101)的流程图。在图15中的步骤S1501中,浏览器(101)根据脚本(401)的执行传输设备类型信息请求。然后,根据多功能外围设备(100)接收设备类型信息请求的事实,开始图20中的处理。在多功能外围设备(100)的CPU 201的控制下执行图20至图24中的处理。

[0077] 在步骤S2001中,多功能外围设备(100)从浏览器(101)中接收设备类型信息请求,并且处理进入步骤S2002。在步骤S2002中,多功能外围设备(100)生成设备类型信息(800),并且处理进入步骤S2003。

[0078] 在步骤S2003中,多功能外围设备(100)将在步骤S2002中生成的设备类型信息(800)传输到浏览器(101)。

[0079] 接下来,将描述图21中的流程图。图21是例示多功能外围设备(100)接收app包(501)并执行app包(501)的安装(步骤S2200)的流程图。

[0080] 在图16中的步骤S1603中,将app包(501)从浏览器(101)传输到多功能外围设备(100)。基于多功能外围设备100接收该app包501,开始图21中的处理。

[0081] 在步骤S2101中,多功能外围设备(100)从浏览器(101)接收app包(501),并将接收到的app包(501)保存在ROM(202)的app程序区域(302)中,并且处理进入步骤S2102。在步骤S2102中,多功能外围设备(100)执行用于安装app包(501)的安装处理(步骤S2200),并且处理进入步骤S2103。下文将描述安装处理(步骤S2200)。在步骤S2103中,多功能外围设备(100)将安装结果传输到浏览器(101)。

[0082] 接下来,将描述图22的流程图。图22是例示由多功能外围设备(100)执行的app包(501)的安装的流程图。该处理对应于图21中的步骤S2102。

[0083] 在步骤S2201中,多功能外围设备(100)在操作单元(204)上显示安装期间画面(图13中的1301),并且处理进入步骤S2202。

[0084] 在步骤S2202中,多功能外围设备(100)检查多功能外围设备(100)接收到的app包(501)的签名(605)以验证app包(501)的有效性,并且处理进入步骤S2203。通过该处理能够检测对接收到的app包(501)的篡改。

[0085] 在步骤S2203中,多功能外围设备(100)检查在步骤S2202中通过多功能外围设备(100)进行的验证的结果是否没有问题。例如,如果在步骤S2202中验证表示在步骤S2202中任何已验证程序都已崩溃或更改,则表示存在问题。如果验证结果没有问题(在步骤S2203

中为“是”),则处理进入步骤S2204。如果验证结果有问题(在步骤S2203中为“否”),则处理进入步骤S2210以停止安装。

[0086] 在步骤S2204中,多功能外围设备(100)检查通过该处理进行的安装(步骤S2200)是否为新安装。如果安装不是新安装(在步骤S2204中为“否”),则处理进入步骤S2207。如果安装是新安装(在步骤S2204中为“是”),则处理进入步骤S2205以确定app的有效状态。

[0087] 在步骤S2205中,多功能外围设备(100)检查已安装的现有目标app是否处于有效状态。如果目标app处于有效状态(在步骤S2205中为“是”),则针对安装需要将app变更为无效状态。因此,处理进入步骤S2206。如果目标app处于无效状态(在步骤S2205中为“否”),则处理进入步骤S2207。

[0088] 在步骤S2206中,多功能外围设备(100)将处于有效状态的目标app变更为无效状态,并且处理进入步骤S2207。

[0089] 在步骤S2207中,多功能外围设备(100)将接收到的app包(501)加载到app程序区域(302)中,并且处理进入步骤S2208。

[0090] 在步骤S2208中,多功能外围设备(100)检查在步骤S2207中进行的加载处理是否成功。例如,如果ROM(202)的app程序区域(302)的可用空间不足,则加载失败(在步骤S2208中为“否”)。因此,处理进入步骤S2210以停止安装。如果加载成功(在步骤S2208中为“是”),则处理进入步骤S2209。

[0091] 在步骤S2209中,参照加载的app包(501)中包括的设置文件(604),多功能外围设备(100)执行app设置处理。下文将描述该app设置处理(图23中的处理)。

[0092] 在步骤S2210中,多功能外围设备(100)在操作单元(204)上显示安装失败画面(1303)并删除app包(501),以通知用户app包(501)在多功能外围设备(100)上的安装失败。

[0093] 接下来,将描述图23中的流程图。图23是图22中步骤S2209的部分。图23是例示由多功能外围设备(100)执行的app设置处理的流程图。此时,多功能外围设备(100)参照app设置文件(604)进行该处理。在本示例性实施例中,将基于app设置文件(604)的示例(700)提供描述。

[0094] 在步骤S2301中,多功能外围设备(100)参照在图7的app设置文件(604)的示例(700)中描述的“uninstallableFlag”(701)的设置值。此时,如果设置为“假”(在步骤S2301中为“假”),则处理进入步骤S2302。如果设置为“真”(在步骤S2301中为“真”),则处理进入步骤S2303。

[0095] 在步骤S2302中,多功能外围设备(100)将app的卸载性能设置为可卸载,并且处理进入步骤S2303。

[0096] 在步骤S2303中,多功能外围设备(100)参照在app设置文件(604)的示例(700)中描述的“runawayDecisionSec”(702)的设置值。此时,如果设置了任意值(在步骤S2303中为“是”),则处理进入步骤S2304。如果没有设置值(在步骤S2303中为“否”),则处理进入步骤S2305。

[0097] 在步骤S2304中,多功能外围设备(100)将“runawayDecisionSec”(702)的设置值设置为app监控时间,并且处理进入步骤S2305。

[0098] 在步骤S2305中,基于安装时自动启动条件(900),多功能外围设备(100)执行自动启动确定(步骤S2400),并且处理进入步骤S2306。下文将描述该自动启动确定(步骤

S2400)。

[0099] 在步骤S2306中,基于步骤S2305的结果(参照图24描述的细节),多功能外围设备(100)确定是否进行针对自动启动app的自动启动处理。此时,如果要自动启动app(在步骤S2306中为“是”),则处理进入步骤S2307。否则,终止该操作的处理。

[0100] 在步骤S2307中,为了自动启动app,多功能外围设备(100)在操作单元(204)上显示空白画面(1302),并且处理进入步骤S2308。

[0101] 在步骤S2308中,多功能外围设备(100)使已安装app有效,并且处理进入步骤S2309。

[0102] 在步骤S2309中,多功能外围设备(100)在操作单元(204)上显示app画面(1304)。

[0103] 接下来,将描述图24中的流程图。图24例示多功能外围设备(100)执行的app自动启动确定的流程。

[0104] 在步骤S2401中,多功能外围设备(100)检查“defaultEnableFlag”(703)的设置。如果设置为“假”(在步骤S2401中为“假”),则多功能外围设备(100)确定自动启动为“否”。如果设置为“真”(在步骤S2401中为“真”),则需要进一步确定,且因此处理进入步骤S2402。

[0105] 在步骤S2402中,多功能外围设备(100)检查当前安装是否为新安装。如果当前安装不是新安装(在步骤S2402中为“否”),则需要进一步确定,且因此处理进入步骤S2403。如果当前安装为新安装(在步骤S2402中为“是”),则多功能外围设备(100)确定自动启动为“是”。

[0106] 在步骤S2403中,多功能外围设备(100)在该安装处理之前检查目标app是否有效。如果目标app不为有效(在步骤S2403中为“否”),则多功能外围设备(100)确定自动启动为“否”。如果目标app为有效(在步骤S2403中为“是”),则多功能外围设备(100)确定自动启动为“是”。

[0107] 此为本示例性实施例的配置。虽然上述示例性实施例中没有描述,但是也可以进行以下控制作为根据本发明的控制方法。

[0108] 例如,从多功能外围设备(100)获取的设备类型信息(800)不仅可以包括设备类型名称(801)和型号信息(802),还可以包括操作单元(204)的尺寸、ROM(202)的容量和剩余容量、DRAM(203)的尺寸和许可证信息。

[0109] 此外,设备类型/app包对应列表(502)可以具有不仅包括设备类型名称(801)还包括DRAM(203)的尺寸、许可证信息和操作单元(204)的尺寸的项目。

[0110] 在多功能外围设备(100)刚一安装之后的情况下,根据图14中处理的执行,在浏览器(101)上显示图10中的画面,并且浏览器(101)在步骤S1403中等待。基于按下了设备连接按钮1004,执行步骤S1404和步骤S1405的处理。然后,在步骤S1406中,浏览器(101)从多功能外围设备(100)获取已安装app包列表(1201)。如果在安装多功能外围设备(100)即刻后没有进行任何安装,则将表示该状态的信息传输到浏览器(101)。接下来,浏览器(101)将在步骤S1405中获取的设备类型信息(800)传输到app服务器装置2001。假设多功能外围设备(100)的设备类型信息(800)的设备类型名称(801)为“STERA\_X3”。app服务器装置2001进行图18中的处理。参照设备类型/app包对应列表(502)的“STERA\_X3”字段,app服务器装置2001如步骤S1802所述生成可安装包列表(1101)。然后,app服务器装置2001将可安装包列表(1101)传输到浏览器(101)。在该示例中,生成包括应用1和应用2的列表并将所述列表传

输到浏览器(101)。

[0111] 基于使用上述信息执行脚本(401),如步骤S1503中所例示而生成app安装管理画面(1100)。在此情况下,在浏览器(101)上显示图11中的画面。然后,基于按下了图11中针对app包的动作(1105)的按钮,进行图16中的处理。在该示例中,浏览器(101)请求app服务器(103)以获取应用1。然后,执行步骤S1602和步骤S1603。最终,将包括应用1的程序体和app设置文件(604)的应用信息从app服务器(103)传输到浏览器(101)。然后,将应用信息从浏览器(101)传输到多功能外围设备(100)。然后,在图17中的步骤S1701中,作为对步骤S1603的响应,获取已安装app包列表(1201)。然后,在步骤S1702中,使用包括已安装app包列表(1201)的信息在网络浏览器(101)上更新显示。即,使用已安装app包列表(1201)作为输入并且基于脚本(401)的执行,将图11中的画面更新为图12中的画面,并且在浏览器(101)上显示图12中的画面。

[0112] 服务器装置的示例为app服务器装置2001。外围设备的示例为多功能外围设备(100)。提供网络浏览器(101)的信息处理装置的示例为客户端PC 2000。

[0113] CPU 2503执行经由网络浏览器(101)接收到的脚本。

[0114] 根据CPU 2503对脚本的执行,从多功能外围设备(100)接收关于多功能外围设备(100)的设备识别信息(识别信息的接收示例)。作为设备类型信息(800)例示设备识别信息的示例。

[0115] 浏览器(101)将接收到的设备类型信息(800)传输到app服务器装置2001。浏览器(101)从app服务器装置2001接收应用信息。

[0116] 浏览器(101)将其接收到的应用信息传输到多功能外围设备(100),以便多功能外围设备(100)安装应用。

[0117] 此外,根据接收应用信息的事实(应用信息的接收示例),进行以下处理。此为在网络浏览器(101)的画面上显示确认画面的处理,所述确认画面包括用于执行脚本和识别应用信息的显示并且所述确认画面还用于发出进行安装的指令。

[0118] 此外,在显示单元的示例DSP 2508显示确认画面之后,根据用户提供预定输入的事实执行脚本。此外,浏览器(101)根据预定输入将应用信息传输到多功能外围设备(100)。

[0119] 应用信息包括应用程序和应用程序的启动条件。此外,多功能外围设备(100)根据启动条件控制应用程序的启动。图7中例示启动条件的示例。

[0120] 应用信息包括表示有效化多功能外围设备(100)上安装的应用的信息。基于应用信息,进行控制以确定是否要在多功能外围设备(100)中使已安装程序有效。

[0121] 应用信息包括用于识别多功能外围设备(100)上安装的应用的信息。此外,根据用于识别多功能外围设备(100)上安装的应用的信息进行卸载。即,根据脚本的执行,在网络浏览器(101)的画面上显示包括指令单元的画面,所述指令单元用于根据用于识别应用的信息从多功能外围设备(100)卸载已安装应用。

[0122] 讨论了作为服务器装置的示例的app服务器装置2001,将所述服务器装置配置为与提供了被配置为与多功能外围设备(100)通信的网络浏览器的客户端PC 2000通信。

[0123] App服务器装置2001将脚本传输到客户端PC 2000。根据客户端PC2000执行了脚本的事实,app服务器装置获取从多功能外围设备(100)获取的关于多功能外围设备(100)的设备类型识别信息。app服务器装置指定对应于获取的设备类型识别信息的应用程序。app

服务器装置2001将包括指定应用程序的应用信息和用于指示多功能外围设备(100)安装应用程序的画面信息传输到客户端PC 2000。

[0124] 根据本发明的示例性实施例,用户不需要预先准备app包。此外,使用用于提供app包和脚本的app提供页,通过单击可以从app服务器获取app包,并且在外围设备上安装并有效化app包。然后,可以在安装app时写入app名称和文件路径,并且每次安装后都使app有效。该配置使用户能够通过单击来安装并使app有效,因此减轻了用户引入app时的负担。

[0125] 根据本示例性实施例,可以提供针对有效地管理外围设备上运行的应用的机制。根据本示例性实施例,用户通过简单操作便可确认外围设备所需的应用,并且使所需应用在外围设备中的使用有效。

[0126] 根据本示例性实施例的一个方面,可以提供针对使用户通过简单操作来确认外围设备所需的应用,并且使所需应用在外围设备中的使用有效的机制。

[0127] 其他实施例

[0128] 本发明的实施例还可以通过如下的方法来实现,即,通过网络或者各种存储介质将执行上述实施例的功能的软件(程序)提供给系统或装置,该系统或装置的计算机或是中央处理单元(CPU)、微处理单元(MPU)读出并执行程序的方法。

[0129] 虽然参照示例性实施例对本发明进行了描述,但是应理解,本公开并不限于所公开的示例性实施例。应当对所附权利要求的范围赋予最宽泛的解释,以使其涵盖所有这些变型例以及等同的结构及功能。

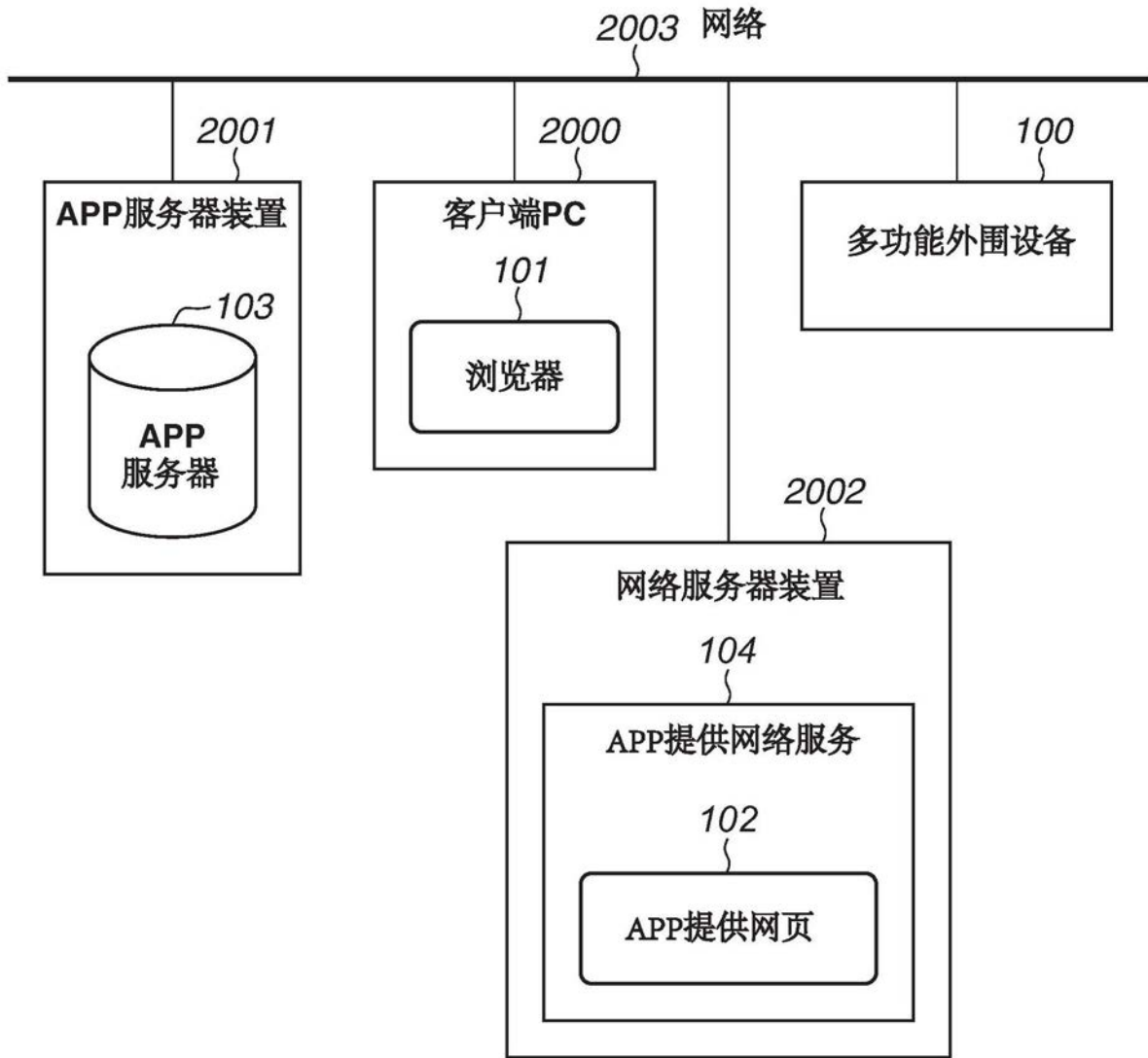


图1

多功能外围设备 (100) 的配置图

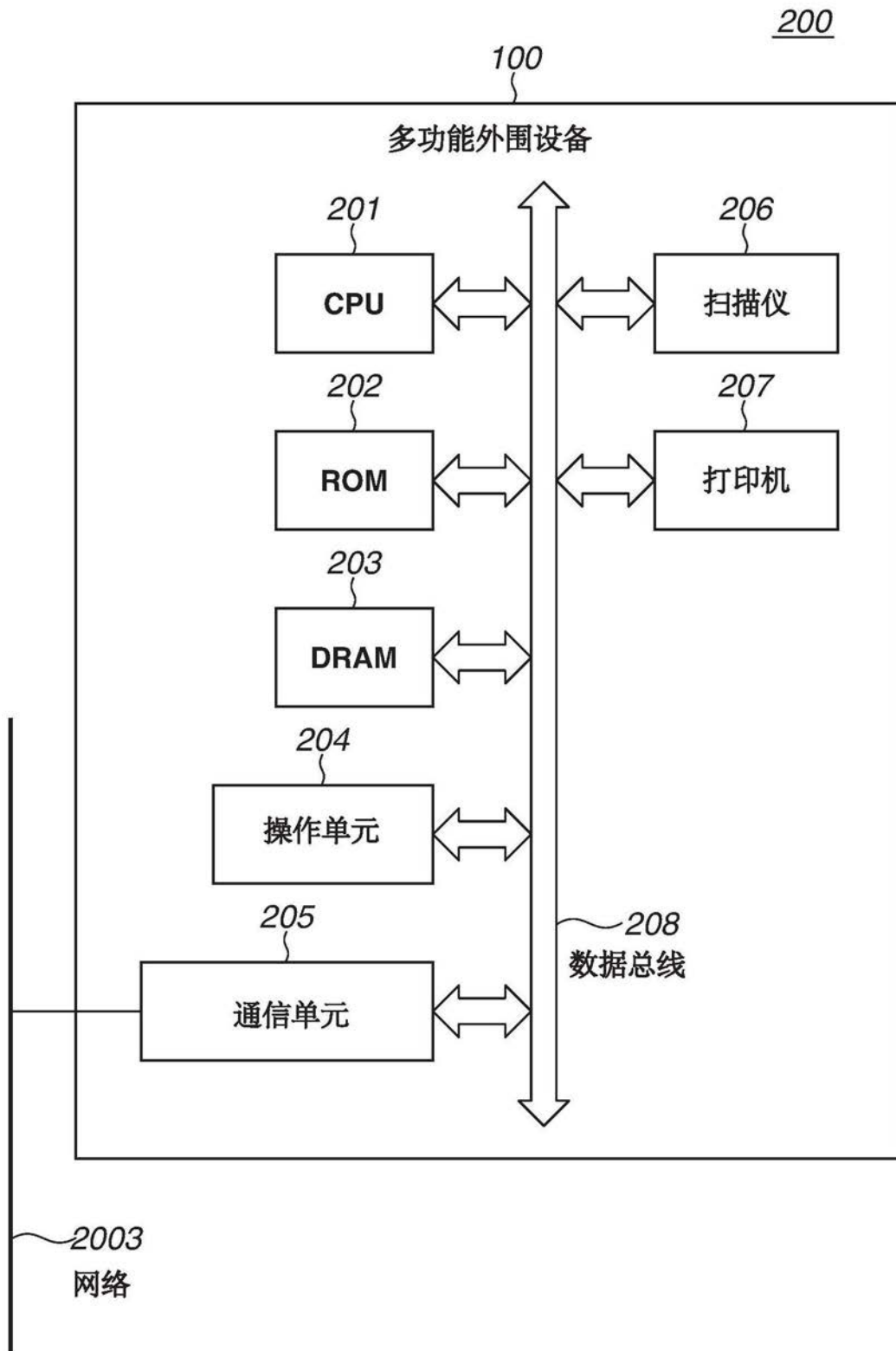


图2

ROM (202) 的配置图

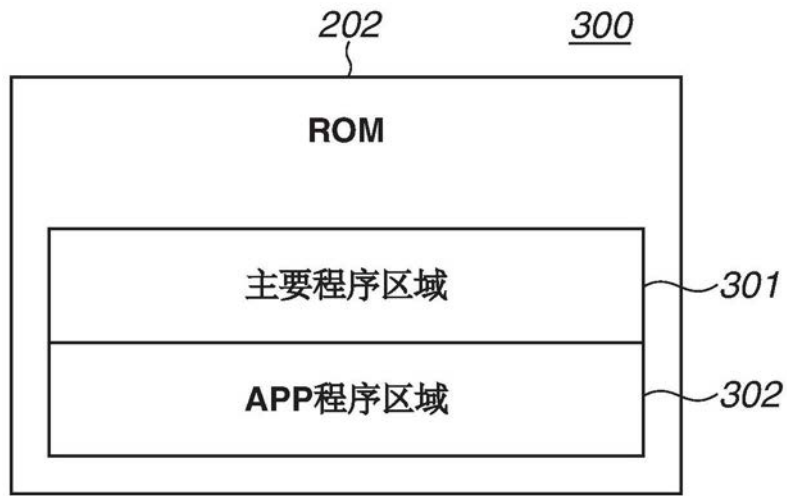


图3

APP提供网页 (102) 的配置图

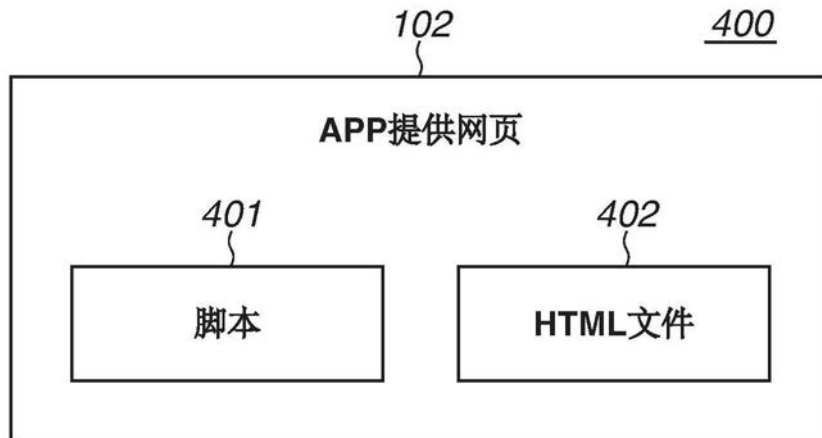


图4

APP服务器 (103) 的配置图

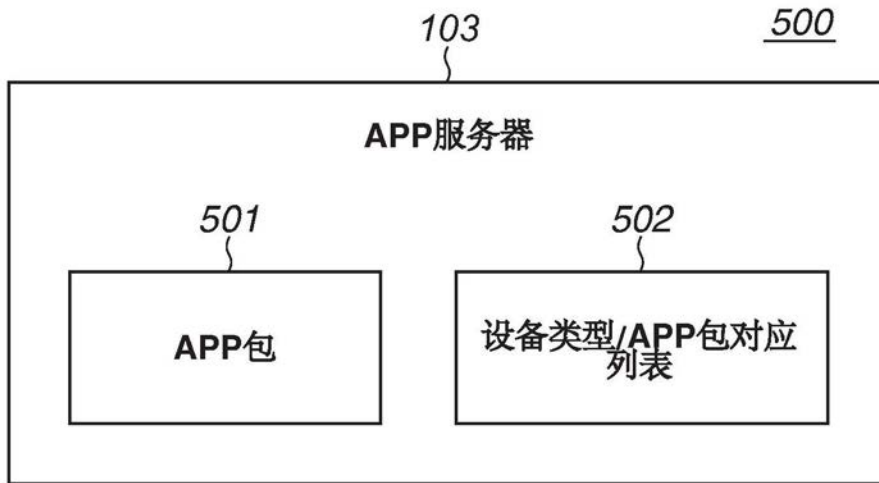
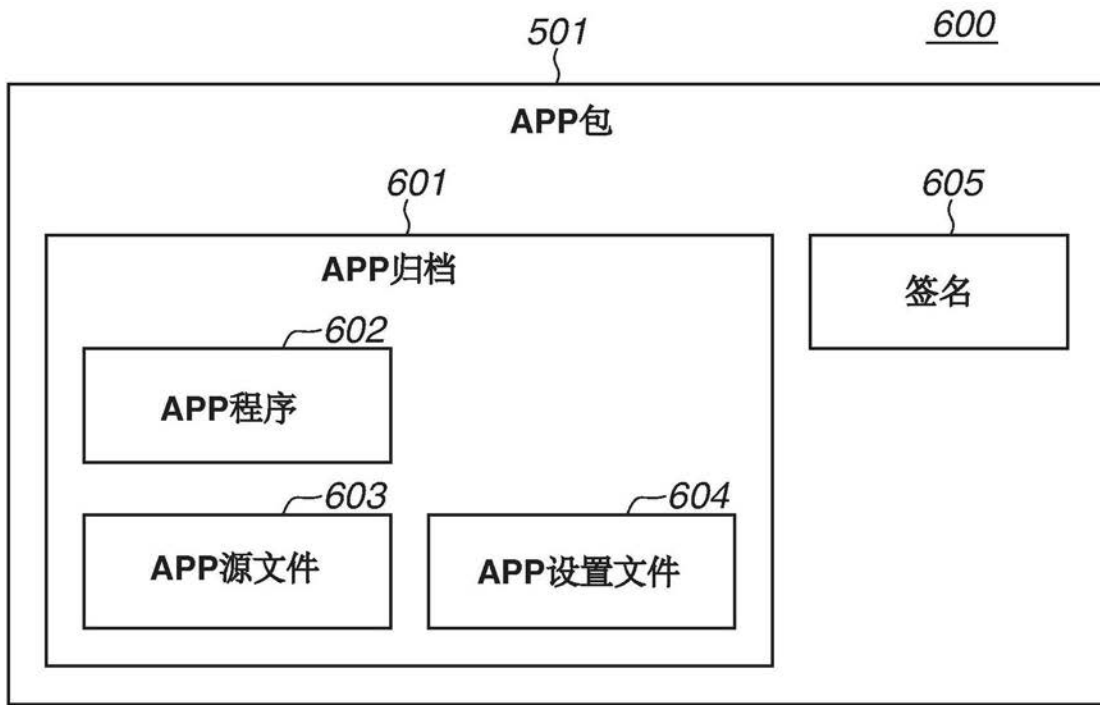


图5

APP包 (501) 的配置图



设备类型/APP包对应列表 (502) 的示例

设备类型名称	APP包
STERA_X1	应用1、应用3、应用5
STERA_X2	应用1、应用5、应用9
STERA_X3	应用1、应用2

图6

APP设置文件的示例

604

设置值名称	设置值
701 ~ <b>uninstallableFlag</b>	假
702 ~ <b>runawayDecisionSec</b>	<b>3500</b>
703 ~ <b>defaultEnableFlag</b>	假
...	...

图7

关于多功能外围设备 (100) 的设备类型信息的示例

800

设备类型信息名称	设备类型信息值
801 ~ 设备类型名称	<b>STERA_XN</b>
802 ~ 型号信息	<b>MODEL_X</b>
...	...

图8

安装时的自动启动条件

900

703~	<b>defaultEnableFlag</b>	新安装	应用状态	自动启动
	假	否	无效	否
			有效	否
		是	任意	否
	真	否	无效	否
			有效	是
		是	任意	是

图9

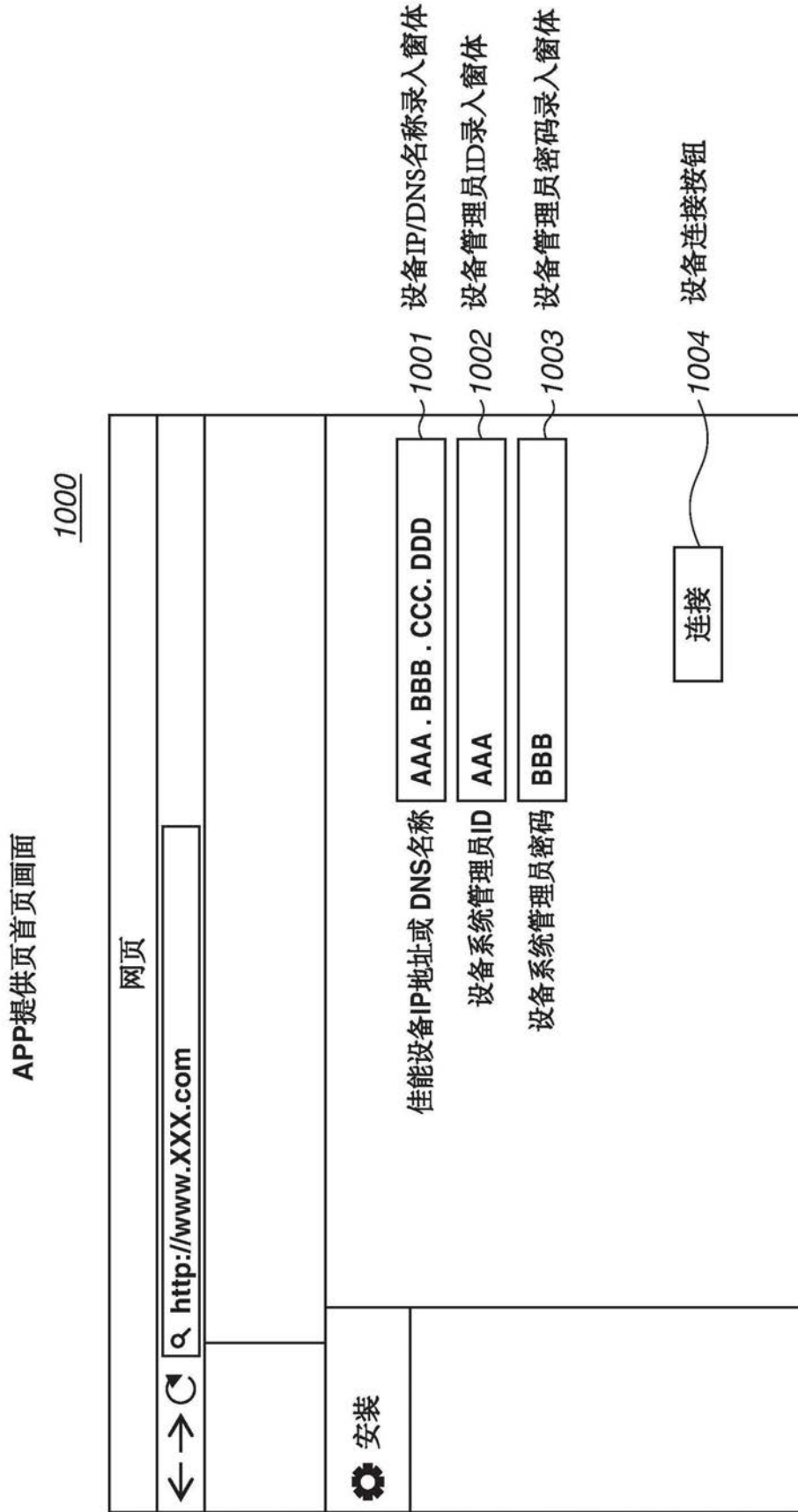


图10

没有已安装APP的情况下的APP 安装管理画面

1100

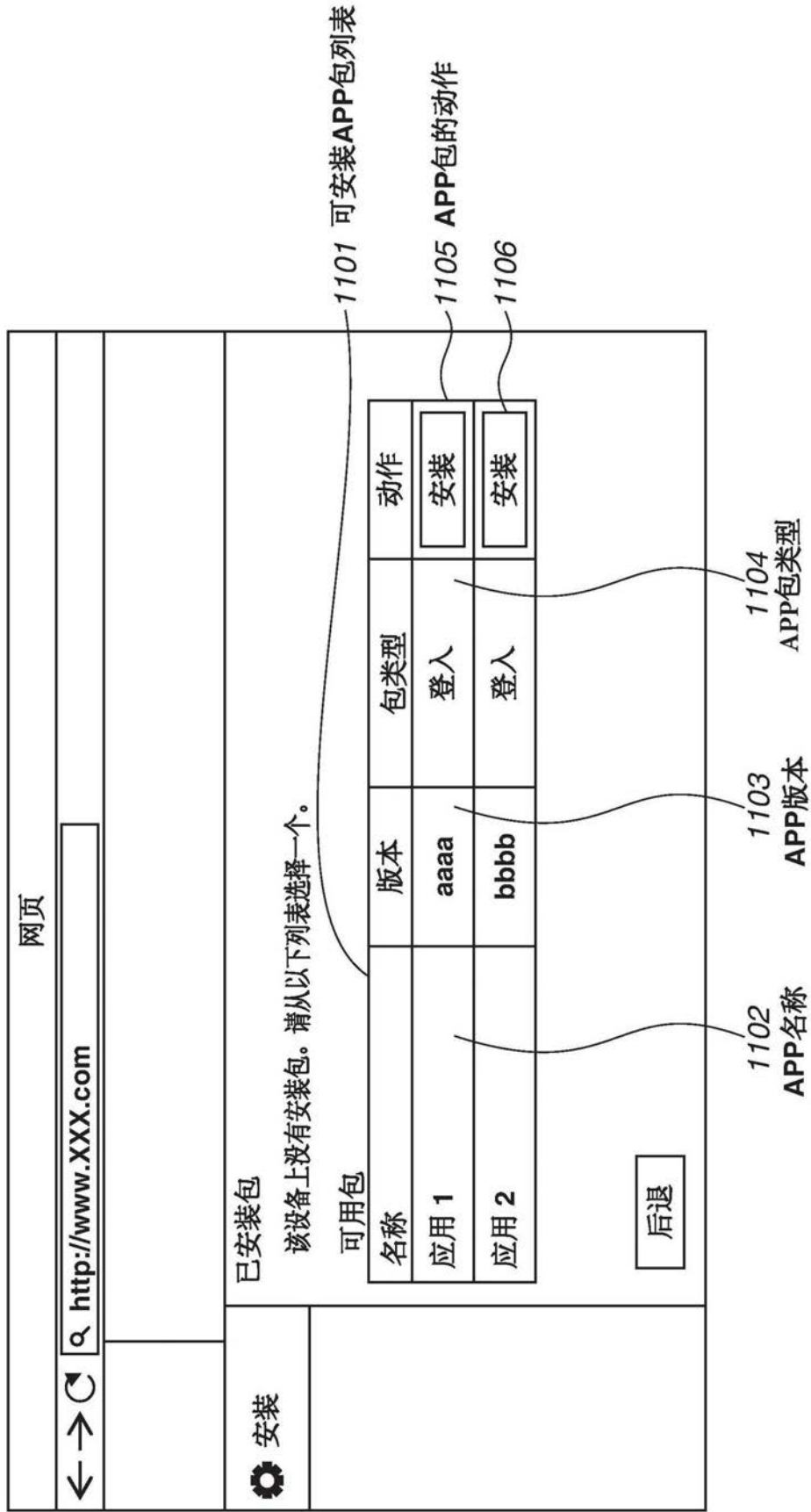


图11

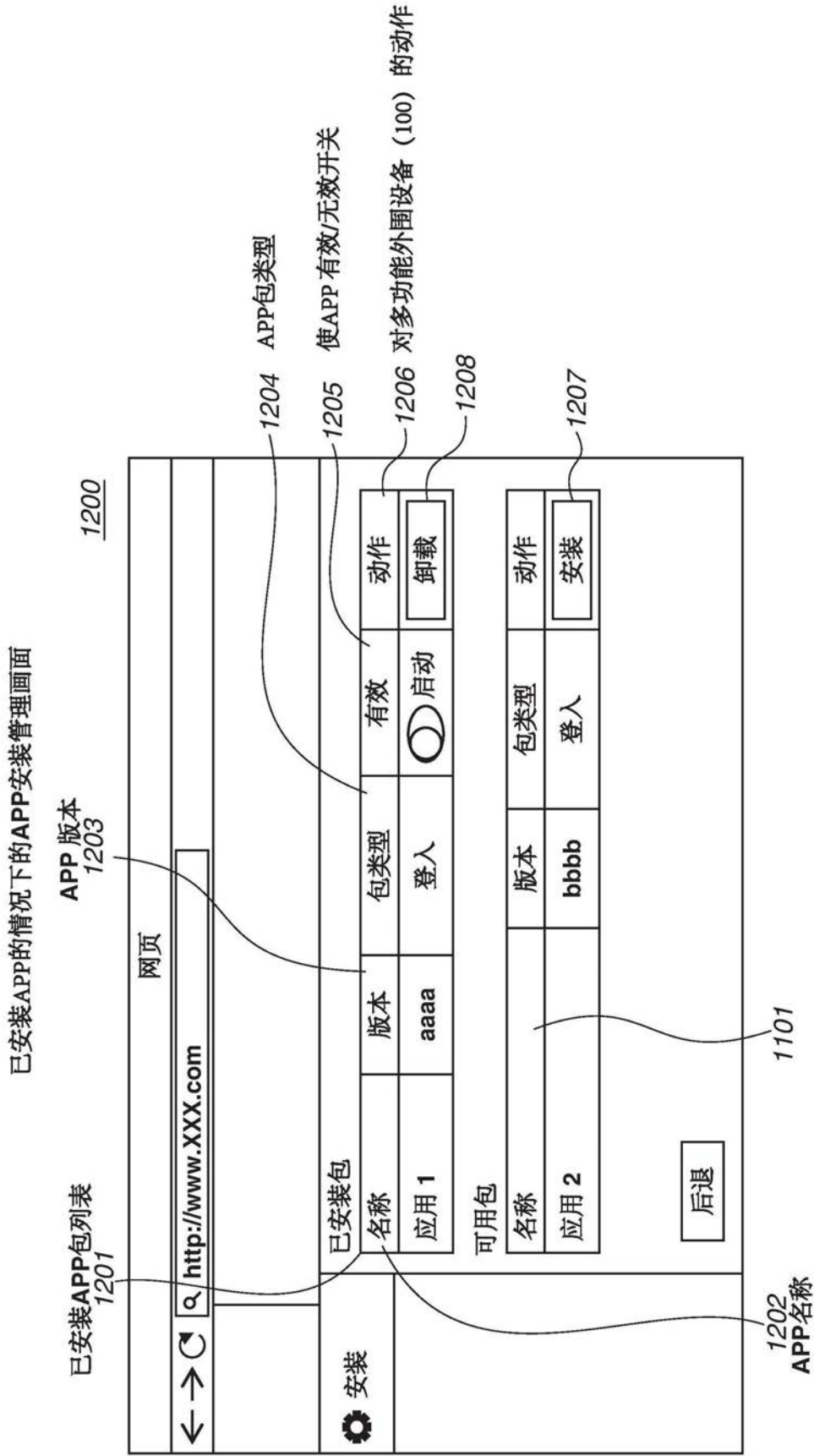


图12



图13

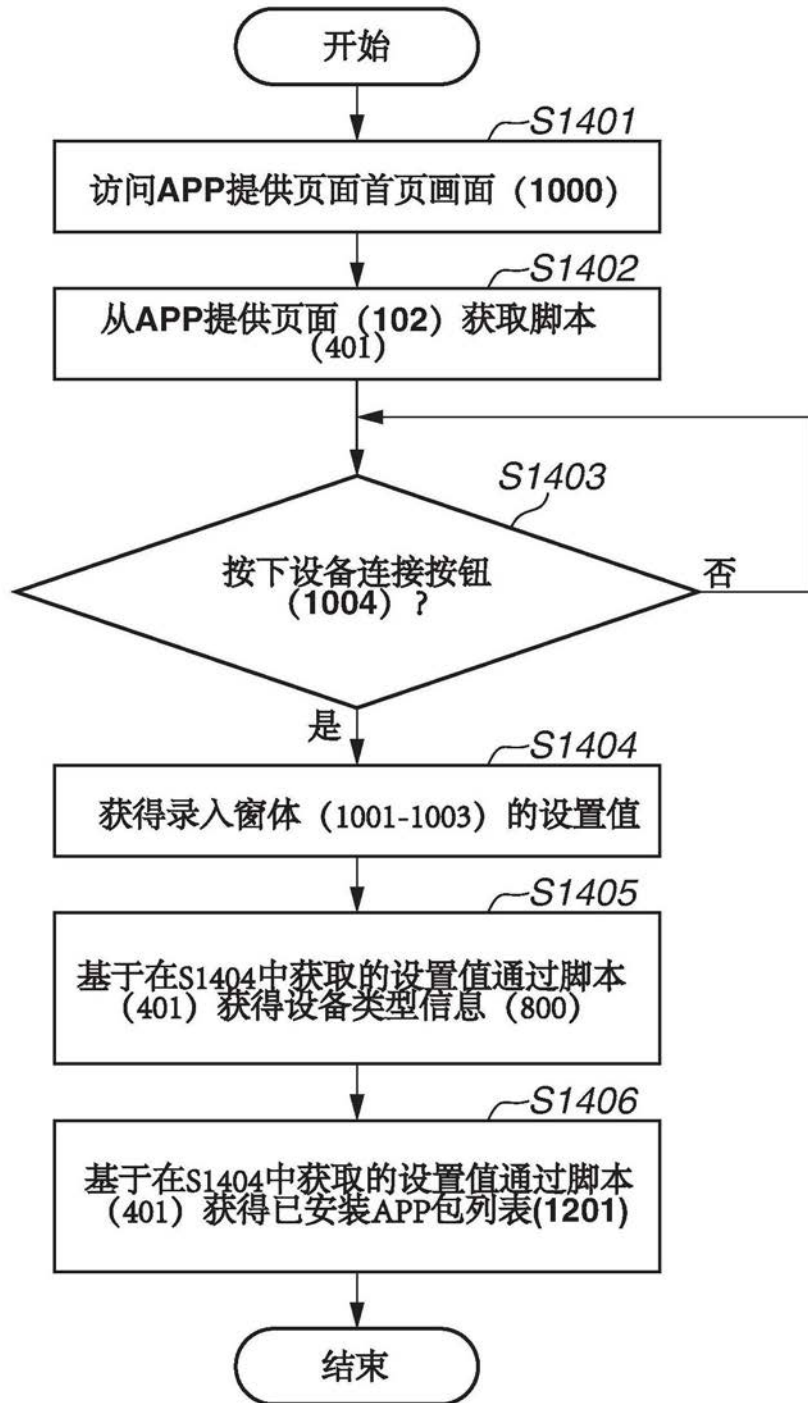


图14

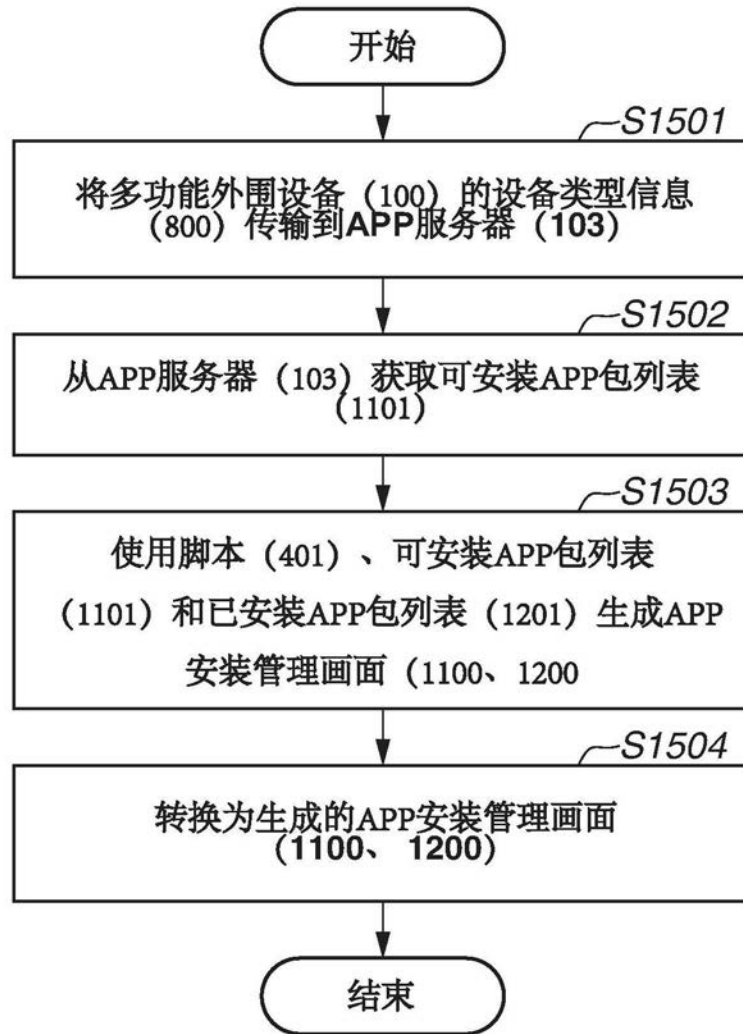


图15

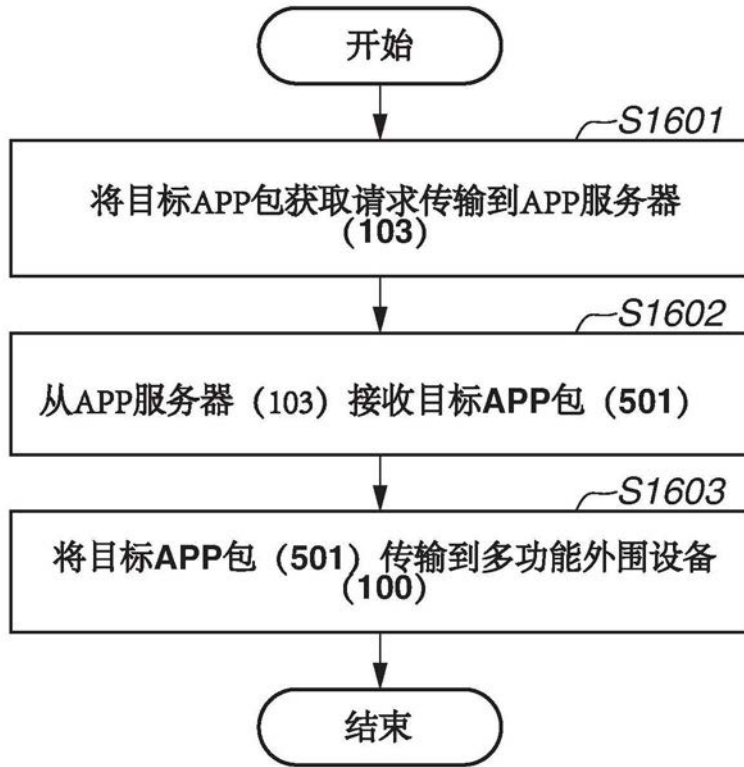


图16

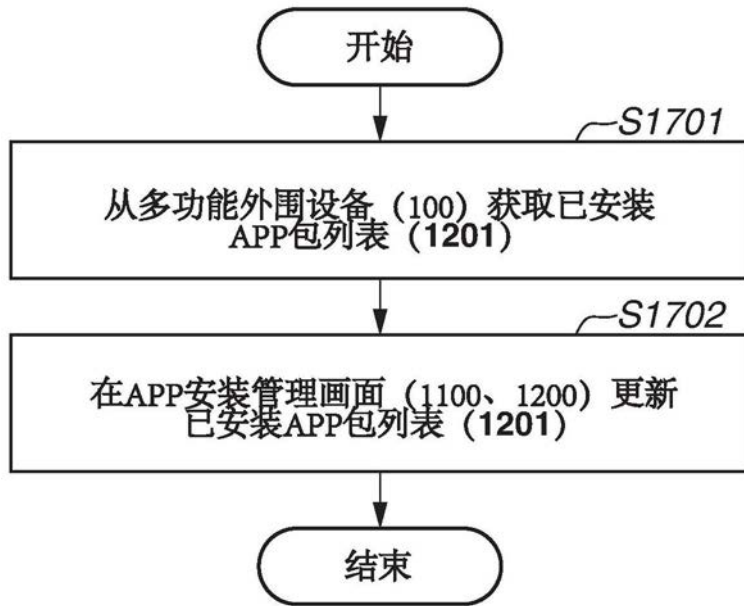


图17

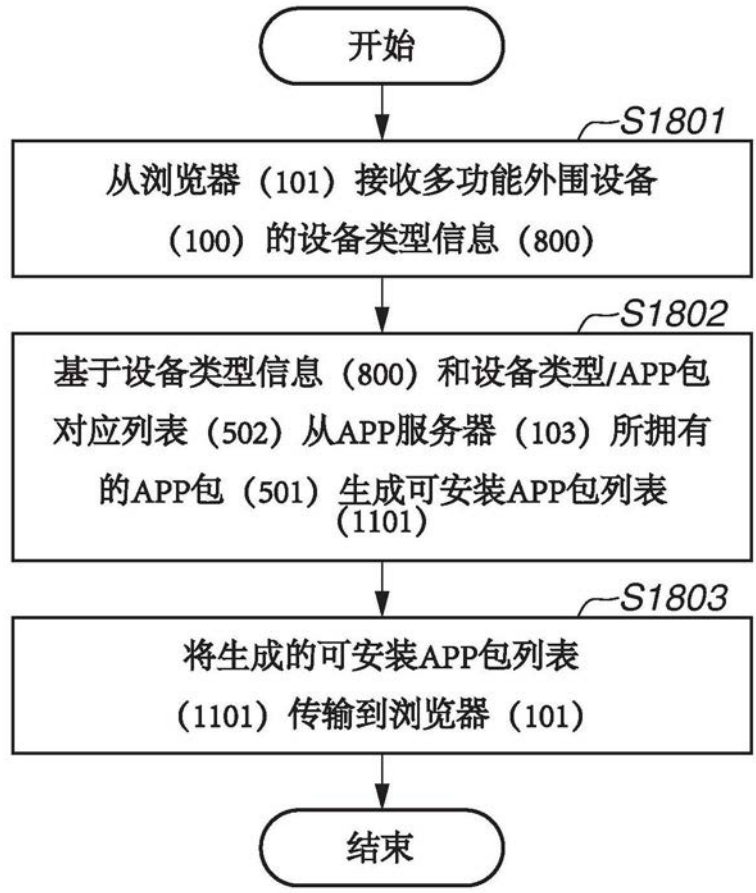


图18

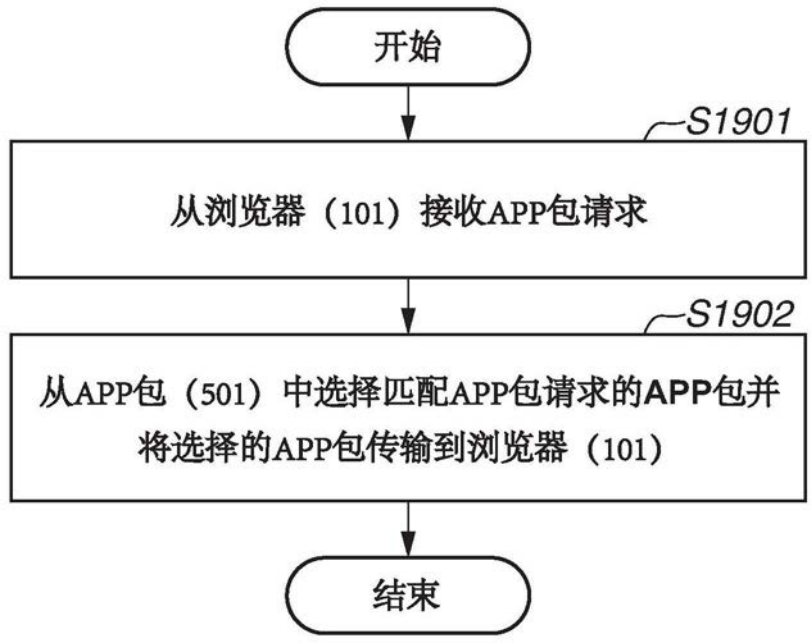


图19

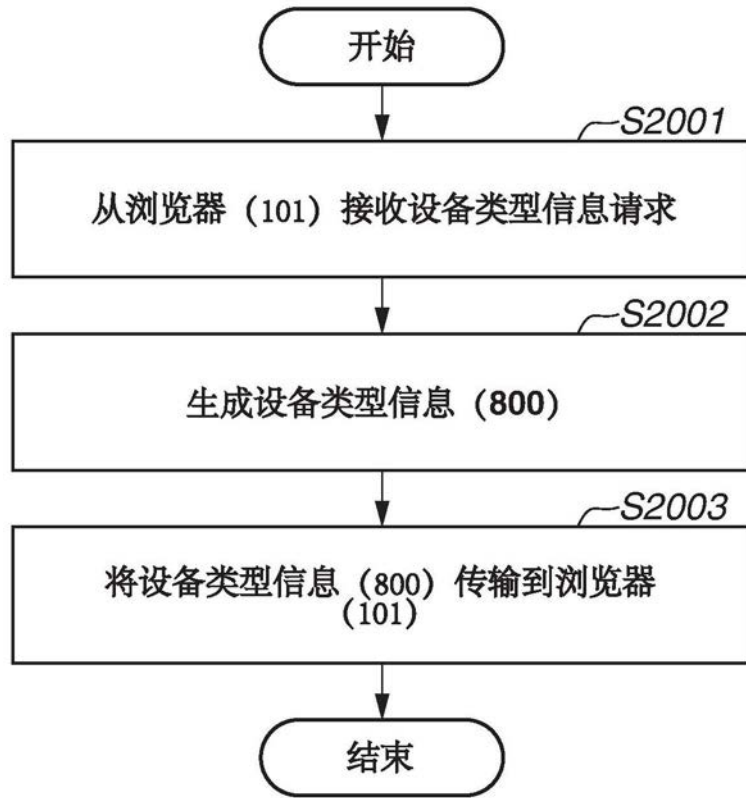


图20

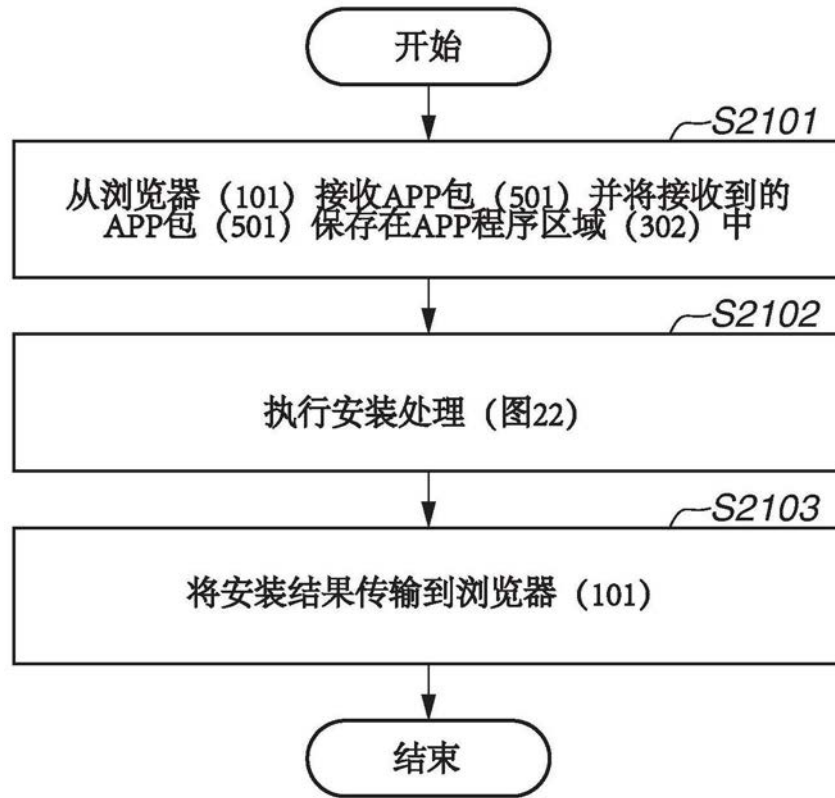


图21

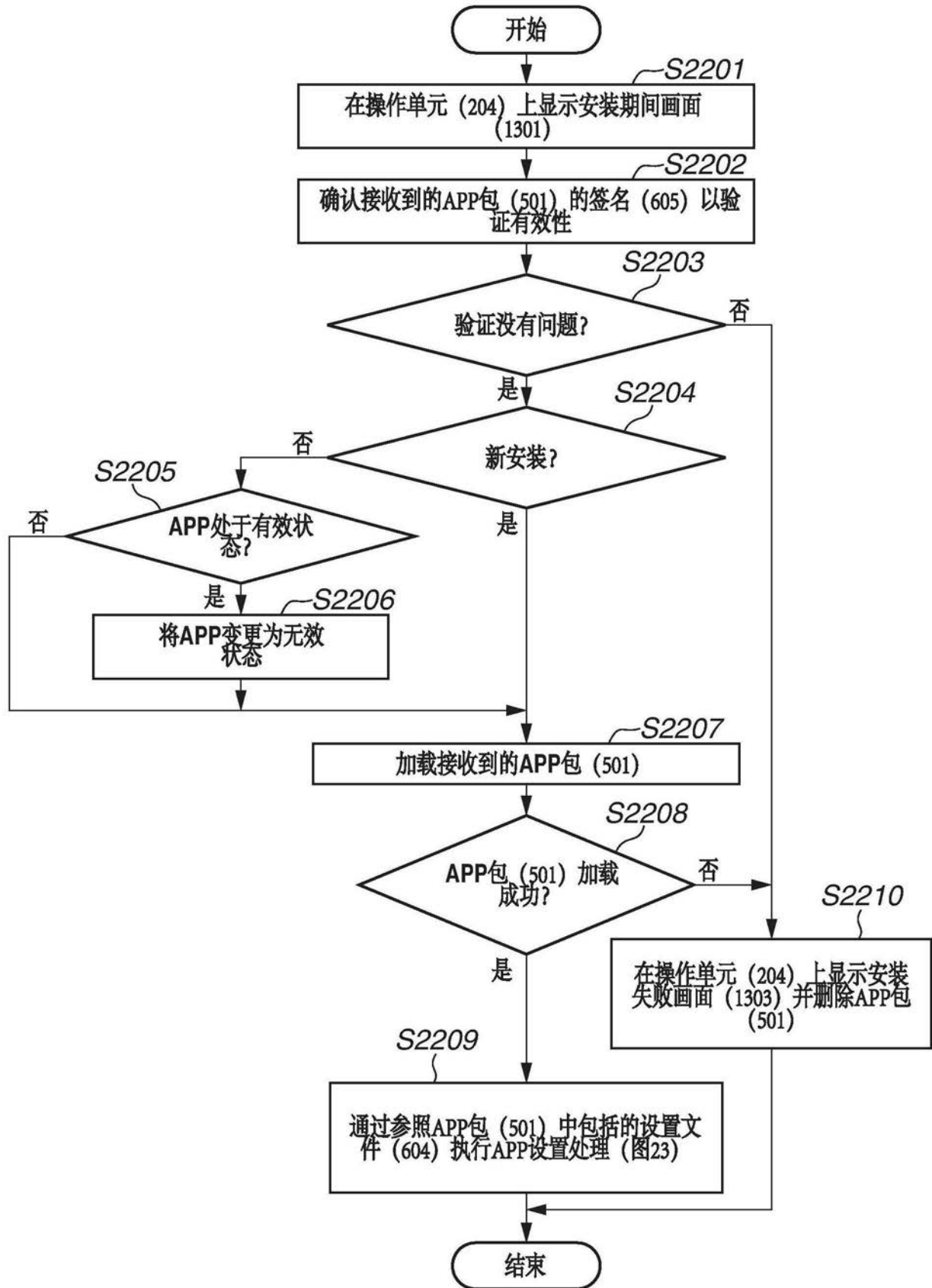


图22

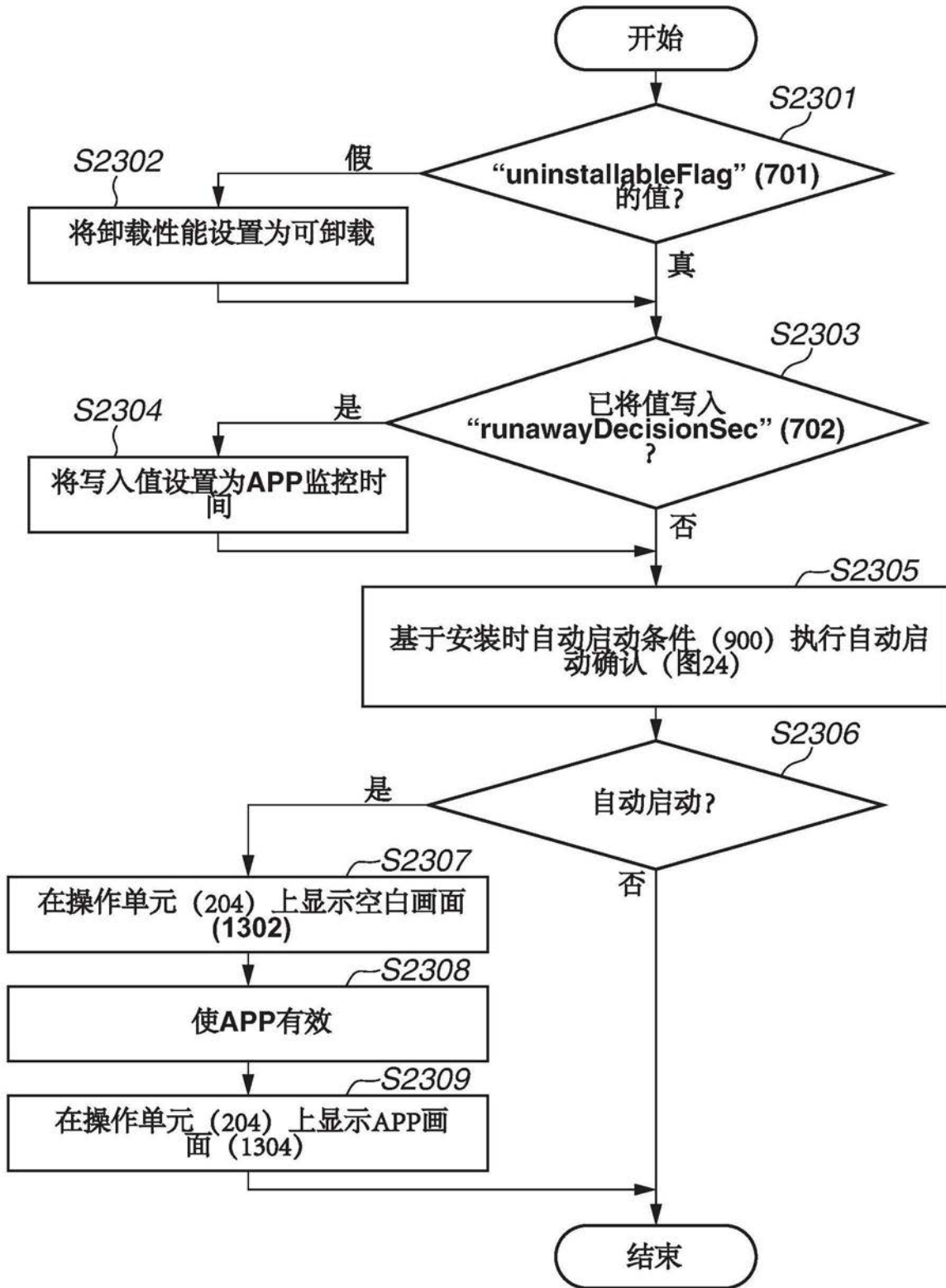


图23

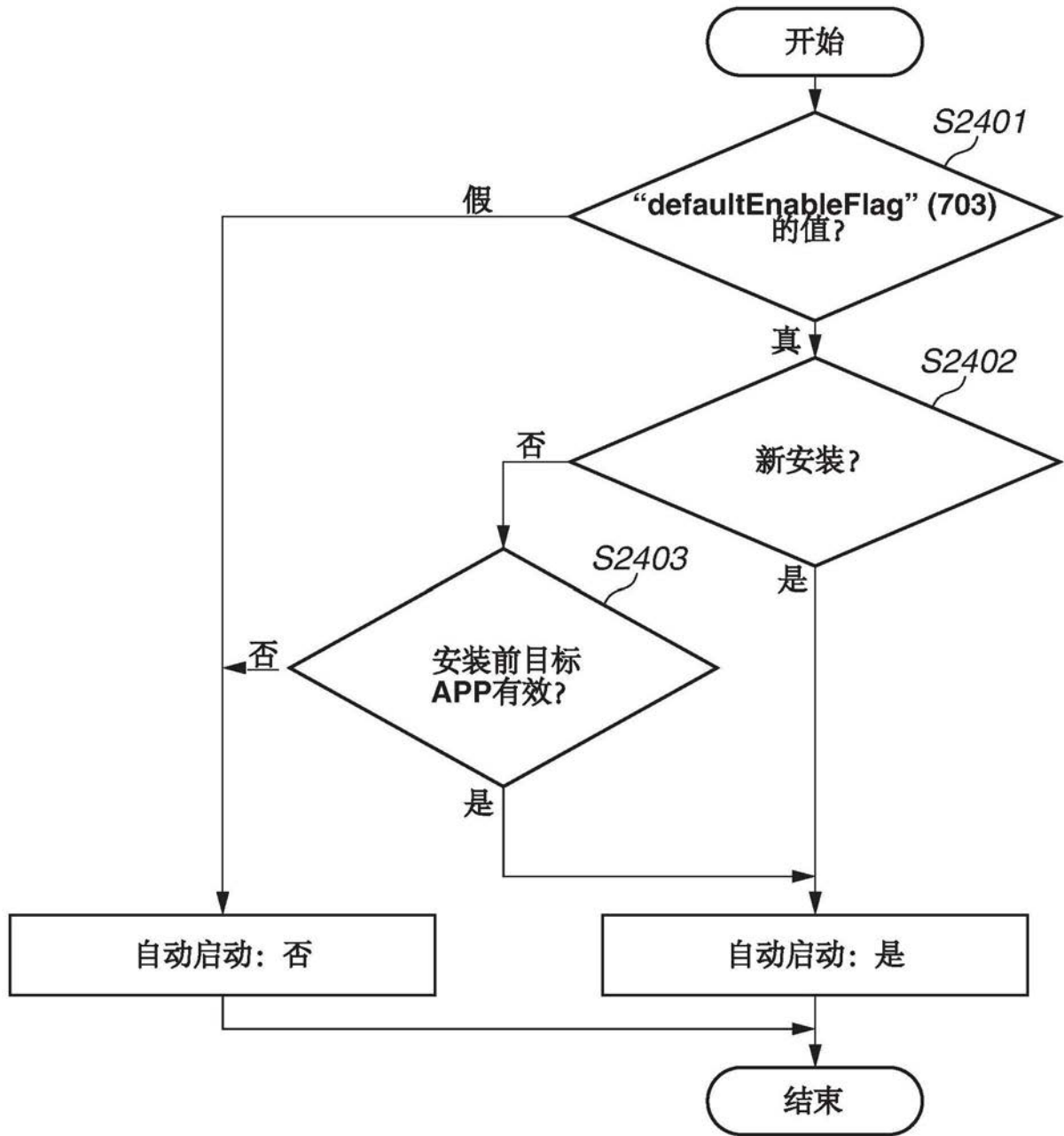


图24

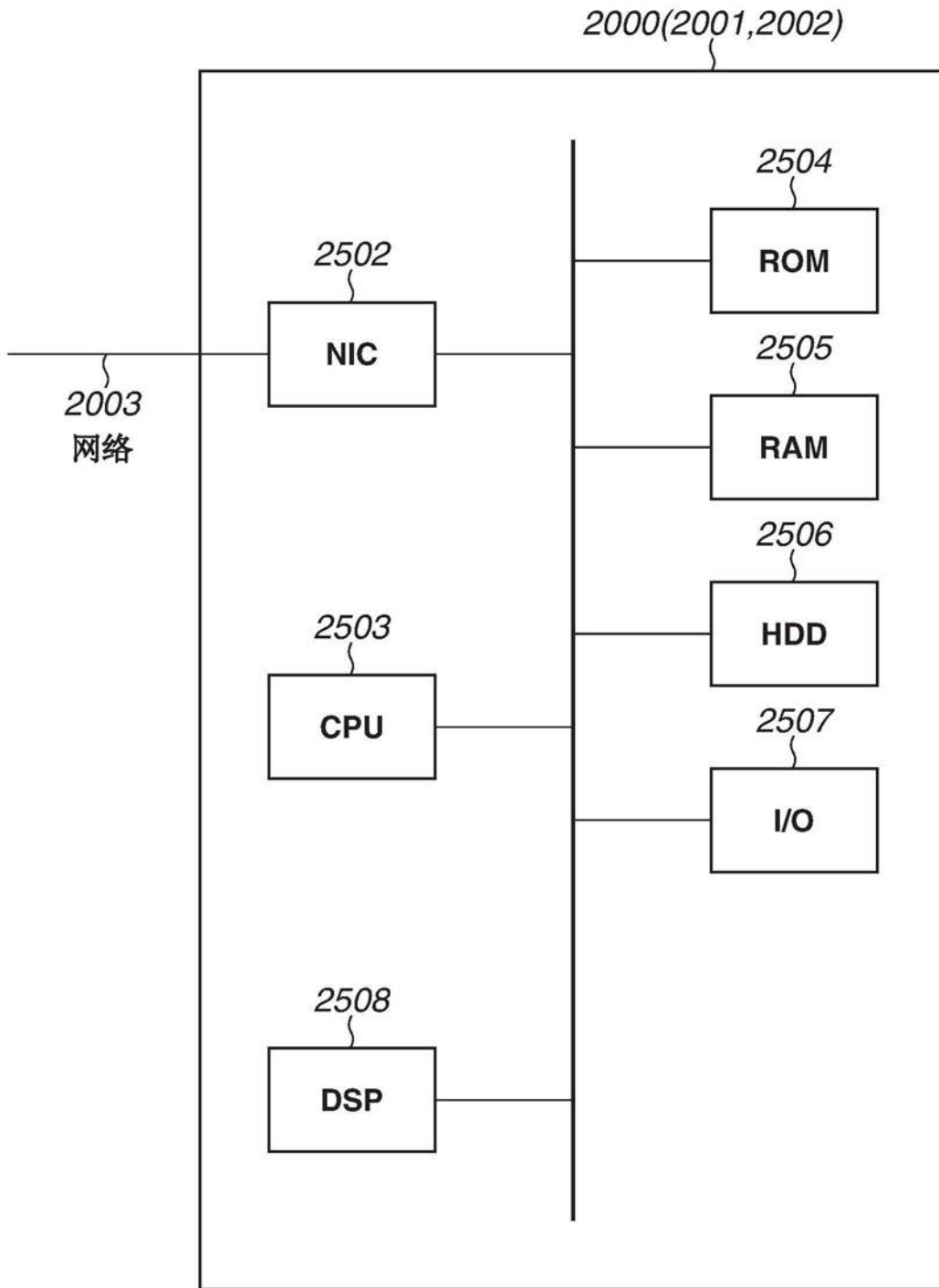


图25