



(12) 发明专利

(10) 授权公告号 CN 101571809 B

(45) 授权公告日 2012.08.22

(21) 申请号 200910142937.1

审查员 张涛

(22) 申请日 2009.05.14

(73) 专利权人 阿里巴巴集团控股有限公司

地址 英属开曼群岛大开曼岛

(72) 发明人 张利明 闻波 周夏衍

(74) 专利代理机构 北京同达信恒知识产权代理

有限公司 11291

代理人 郭润湘

(51) Int. Cl.

G06F 9/445 (2006.01)

H04L 29/06 (2006.01)

(56) 对比文件

CN 101291305 A, 2008.10.22, 全文.

CN 1524216 A, 2004.08.25, 全文.

CN 1968154 A, 2007.05.23, 说明书第5页第

1、4段、第5页第9段-第6页第2段.

CN 101303650 A, 2008.11.12, 全文.

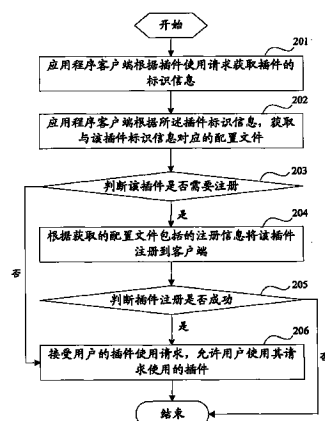
权利要求书 1 页 说明书 8 页 附图 3 页

(54) 发明名称

一种插件注册的实现方法及其装置

(57) 摘要

本申请公开了一种插件注册的实现方法及其装置,具体技术方案包括:应用程序客户端根据插件使用请求获取插件标识信息;应用程序客户端根据所述插件标识信息,获取与所述插件标识信息对应的配置文件;根据所述配置文件包括的注册信息注册所述插件。采用该技术方案,由应用程序客户端根据预设的配置文件执行插件的注册过程,因此,插件无需自带安装包,并且插件除了适用于对插件是否注册无特殊要求的应用程序,也适用于要求插件必须注册的应用程序,从而提高了插件的适用范围。



1. 一种插件注册的实现方法,其特征在于,包括:

应用程序客户端根据插件使用请求获取插件标识信息;

应用程序客户端根据所述插件标识信息,从所述应用程序客户端预先下载到本地的配置文件中获取与所述插件标识信息对应的配置文件,或从服务器获取与所述插件标识信息对应的配置文件;

应用程序客户端根据所述配置文件包括的注册信息将所述插件注册到系统;其中,所述注册信息包括如下注册项信息中的一个或多个:注册表信息、环境变量信息、公共运行库信息以及插件自定义安装文件信息。

2. 如权利要求1所述的实现方法,其特征在于,所述应用程序客户端根据所述插件标识信息,获取与所述插件标识信息对应的配置文件之前还包括:

服务器接收插件开发者在注册所述插件时提供的与所述插件对应的配置文件;或,

服务器获取所述插件的属性信息,并根据所述属性信息以及所述插件的运行环境要求生成与所述插件对应的配置文件。

3. 如权利要求1或2所述的实现方法,其特征在于,所述应用程序客户端根据所述配置文件包括的注册信息注册所述插件,包括:

确定所述注册信息中指定的注册项;

若所述注册项为一个,则执行所述注册项对应的注册过程;若所述注册项为多个,则依次执行所述注册项分别对应的注册过程。

4. 如权利要求1或2所述的实现方法,其特征在于,所述配置文件还包括:插件对应的运行库信息;

所述方法还包括:根据所述运行库信息,获取对应的运行库安装文件并安装。

5. 如权利要求1或2所述的实现方法,其特征在于,所述插件使用请求为插件安装请求或插件加载请求。

6. 一种用于实现插件注册的应用程序客户端,其特征在于,包括:

接收单元,用于根据插件使用请求获取插件标识信息;

配置文件获取单元,用于根据所述插件标识信息,从所述应用程序客户端预先下载到本地的配置文件中获取与所述插件标识信息对应的配置文件,或从服务器获取与所述插件标识信息对应的配置文件;

注册单元,用于根据所述配置文件包括的注册信息将所述插件注册到系统;其中,所述注册信息包括如下注册项信息中的一个或多个:注册表信息、环境变量信息、公共运行库信息以及插件自定义安装文件信息。

7. 如权利要求6所述的应用程序客户端,其特征在于,所述注册单元,包括:

注册项确定模块,用于确定所述注册信息中指定的注册项;

注册执行模块,用于若所述注册项为一个,则执行所述注册项对应的注册过程;若所述注册项为多个时,则依次执行所述注册项分别对应的注册过程。

8. 如权利要求6所述的应用程序客户端,其特征在于,所述配置文件还包括:插件对应的运行库信息;

所述注册单元还用于,根据所述运行库信息,获取对应的运行库安装文件并安装。

一种插件注册的实现方法及其装置

技术领域

[0001] 本申请涉及通信技术领域,尤其涉及一种插件注册的实现方法及其装置。

背景技术

[0002] 随着插件技术的迅速发展以及插件带来的各种辅助功能,插件的使用已经非常普及,一般软件都带有插件以执行其辅助功能,例如在即时通讯系统中,即时通讯系统客户端通过调用插件程序,以执行特定操作或处理特定类型的文件。目前,插件根据使用时是否需要注册来划分,主要有如下两类:

[0003] 第一类是以自行提供安装包(setup)方式发布的插件,用户在使用此类插件时,首先需要根据插件中自带的安装包执行注册过程,并且在用户初次使用时,还需要执行安装过程;

[0004] 第二类是以免注册方式发布的插件,从而实现插件的完全自包含。用户在使用此类插件时,无需执行插件注册过程,例如,不对注册表进行操作、不对系统敏感区进行操作、不向非自身所在目录外的目录进行写操作等,并且此类插件无需执行安装、删除或卸载等操作。

[0005] 但在实际使用过程中,上述两类插件有如下缺陷:

[0006] 第一类插件需要制作安装包,对于插件开发者而言,由于其无法预知目标客户端(即使用插件的客户端)的客户端环境,因此为了使插件尽可能多地在各种类型的客户端环境中运行,插件自带的安装包需要携带尽可能多的功能,因此安装包尺寸较大,这对用户的下载及安装都造成负担,并且对插件开发者来说,制作安装包本身也是一个非常繁琐的过程;

[0007] 第二类插件相对于第一类来说,由于无需自带安装包,为开发者以及用户提供了方便,但是这类插件的运用场合非常有限,因为很多插件应用场景都需要设置注册信息或者环境变量等,例如,插件需要和平台中其它插件进行交互或者需要被浏览器内的脚本调用等情形下,此类插件都无法适用。

[0008] 综上所述,现有插件都存在局限性,在适用范围广的情况下,插件需要自带安装包,插件开发者需要进行安装包配置过程,并且该安装包尺寸往往较大,给开发者以及用户都造成负担;在无需自带安装包的情况下,插件的适用范围非常有限。

发明内容

[0009] 有鉴于此,本申请实施例提供一种插件注册的实现方法及其装置,采用该技术方案,插件无需自带安装包,并且提高了插件的适用范围。

[0010] 本申请实施例通过如下技术方案实现:

[0011] 本申请实施例提供了一种插件注册的实现方法,包括:

[0012] 应用程序客户端根据插件使用请求获取插件标识信息;

[0013] 应用程序客户端根据所述插件标识信息,从所述应用程序客户端预先下载到本地

的配置文件中获取与所述插件标识信息对应的配置文件,或从服务器获取与所述插件标识信息对应的配置文件;

[0014] 应用程序客户端根据所述配置文件包括的注册信息将所述插件注册到系统;其中,所述注册信息包括如下注册项信息中的一个或多个:注册表信息、环境变量信息、公共运行库信息以及插件自定义安装文件信息。

[0015] 本申请实施例还提供了一种用于实现插件注册的应用程序客户端,包括:

[0016] 接收单元,用于根据插件使用请求获取插件标识信息;

[0017] 配置文件获取单元,用于根据所述插件标识信息,从所述应用程序客户端预先下载到本地的配置文件中获取与所述插件标识信息对应的配置文件,或从服务器获取与所述插件标识信息对应的配置文件;

[0018] 注册单元,用于根据所述配置文件包括的注册信息将所述插件注册到系统;其中,所述注册信息包括如下注册项信息中的一个或多个:注册表信息、环境变量信息、公共运行库信息以及插件自定义安装文件信息。

[0019] 通过以上技术方案,本申请实施例中,应用程序客户端接收用户的插件使用请求后,根据该插件使用请求获取插件标识信息,并进一步获取与该插件标识信息对应的配置文件,并根据该配置文件包括的注册信息注册该插件。采用该技术方案,由应用程序客户端根据预设的配置文件执行插件的注册过程,因此,插件无需自带安装包,从而省去了插件开发者对安装包的制作过程,减轻了插件开发者以及用户的负担,并且采用该技术方案,插件除了适用于对插件是否注册无特殊要求的应用程序,也适用于要求插件必须注册的应用程序,从而提高了插件的适用范围。

[0020] 本申请的其它特征和优点将在随后的说明书中阐述,并且,部分地从说明书中变得显而易见,或者通过实施本申请而了解。

附图说明

[0021] 图1为本申请实施例提供的服务器、客户端、插件之间的逻辑关系图;

[0022] 图2为本申请实施例提供的插件注册的实现方法流程图;

[0023] 图3为本申请实施例提供的客户端结构示意图。

具体实施方式

[0024] 为了给出在插件无需自带安装包的情况下提高插件适用范围的实现方案,本申请实施例提供了一种插件注册的实现方法,该方法中,应用程序客户端接收用户的插件使用请求后,根据该插件使用请求获取插件标识信息,并进一步获取与该插件标识信息对应的配置文件,并根据获取的配置文件包括的注册信息注册该插件。

[0025] 一个较佳的实施例中,配置文件可以由客户端预先从指定服务器中下载并存储在客户端本地,在使用时直接从客户端本地获取,也可以直接存储在网络侧指定的服务器中,在使用时实时下载。

[0026] 一个较佳的实施例中,根据注册信息将用户请求使用的插件注册到系统时,首先确定该注册信息中指定的注册项;若注册项为一个,则执行该注册项对应的注册过程;若注册项为多个,则依次执行各注册项分别对应的注册过程。

[0027] 一个较佳的实施例中,配置文件还包括插件对应的运行库信息;应用程序客户端还需要根据该运行库信息,获取对应的运行库安装文件并安装。

[0028] 一个较佳的实施例中,插件对应的配置文件可以由插件开发者生成,也可以由网络侧服务器根据插件的属性信息以及运行环境要求生成。

[0029] 以下结合说明书附图对本申请的优选实施例进行说明,应当理解,此处所描述的优选实施例仅用于说明和解释本申请,并不用于限定本申请。

[0030] 在执行插件注册过程之前,首先结合插件具体的应用要求确定该插件注册时所需要的注册信息,例如,对于一个客户端应用,其需要的注册信息往往体现为注册表信息以及依赖信息。

[0031] 再具体地,例如图片编辑器插件,其采用 C# 语言开发,那么该插件需要的注册信息往往包括:此插件组件的信息,这些信息通常位于注册表,以及 Net Framework 运行库信息;在例如网页型应用插件,其需要的注册信息一般体现为网页 Javascript 运行所需要的信息,如依赖组件的注册信息,这些信息一般写于注册表,如 ActiveX 控件注册信息。

[0032] 插件的注册信息可以以配置文件的方式保存,一个插件对应一个配置文件。较佳地,插件对应的配置文件中除了存储插件运行所需要的注册信息外,还可以存储插件自身的属性信息。一个具体实施例中,配置文件可以以 XML 文件形式生成,以便于扩展,例如:

[0033]

```
<?xml version="1.0" encoding="utf-8"?>
<Plugin>
  <AppID>4462</AppID>
  <Name>WWNotepad</Name>
  <Version>1.0</Version>
  <WWSDK_Version>3100</WWSDK_Version>

<Manifest>
  <Reg>
```

[0034]

```

    <Name>wwnotepad.dll</Name>
    <CLSID>{1924C837-AC7E-4A76-B92D-007CCBC4296D}</CLSID>
    <ThreadingModel>Apartment</ThreadingModel>
  </Reg>
  <Runtime>
    <Name>.Net FrameWork</Name>
    <Version>3.5</Version>
    <CodeBase>www.alisoft.com/download/dotnetframework_3.5.aspx
  </CodeBase>
</RunTime>
<File>
  <Name>privatefile.dll</Name>
  <Version>1.1</Version>
  <CodeBase>www.myselfwebsite.com/download/privatefile_1.1.aspx</Code
base>
</File>
</Manifest>
</Plugin>

```

[0035] 以上所示配置文件包括插件的属性信息以及注册信息，各字段代表的含义如下表所示：

[0036]

字段名称	含义	备注
AppID	插件 ID	必选
Name	插件名称	必选
Version	插件版本	
WWSDK Version	要求宿主程序的最低版本号	必选
Manifest	注册信息节开始标记	
Reg	写入到注册表的信息	
RunTime	需要的运行库	
File	需要的依赖文件	

[0037] 为插件生成配置文件时可以依据上表中定义的字段以及各字段的含义，结合插件的属性信息以及具体的运行环境要求，利用 XML 语言生成配置文件。

[0038] 一个实施例中，为了方便查找配置文件，在保存配置文件时，建立插件标识信息与配置文件的对应关系，以各配置文件对应的插件标识信息为索引项建立存储路径目录。

[0039] 其中，插件的插件标识信息用于唯一地标识该插件，由服务器分配。服务器为插件

分配标识可以有多种方式,例如,插件开发者在开发插件后,需要注册该插件并提交插件相关信息,例如插件名称等,服务器可以在插件开发者注册插件时为该插件产生一个标识,如 AppID。

[0040] 进一步地,插件标识信息还可以为插件的名称,或者插件的其它属性信息。进一步地,也可以将插件的至少两个属性信息作为该插件的插件标识信息,以在插件种类繁多的情况下,保证插件的插件标识信息的唯一性。

[0041] 本申请一个实施例中,服务器提供公用运行库资源,例如 dotnetframework, java runtime 等公共运行库,以供客户端下载,并且生成各插件对应的配置文件(含插件的注册信息)并存储,以供客户端查询和下载。客户端可以根据用户对插件的使用需求从服务器中获取该插件的配置文件,根据获取的配置文件完成该插件的注册过程。具体如图 1 所示,插件的开发商提供插件的相关属性信息给服务器,服务器为该插件生成配置文件以及公共运行库等信息,客户端可以从服务器获取插件的配置文件等相关信息,并协助完成插件的注册。

[0042] 本申请又一实施例中,插件的配置文件也可以由插件开发者生成,并在注册插件时上传到服务器中,由服务器统一存储,并根据客户端的下载请求提供给客户端。

[0043] 进一步地,为了加快插件的注册速度,客户端可以预先将各插件的配置文件下载到本地,当需要注册时直接从本地获取配置文件,以加快响应速度。并且预先下载到客户端本地的配置文件可以根据需要进行更新。

[0044] 以上所述的配置文件可以存储在客户端本地,也可以存储在服务器中。

[0045] 如图 2 所示,插件注册的实现方法,包括如下步骤:

[0046] 步骤 201、应用程序客户端根据插件使用请求获取插件的标识信息。

[0047] 该步骤中,用户向当前登录的应用程序客户端发送插件使用请求,该插件使用请求可以为插件安装请求,也可以为插件加载请求,或者根据需要,可以是任何调用插件的请求。并且,应用程序客户端可以根据用户的使用请求获取插件的标识信息,一般来说用户请求使用某个插件时,不管是体现在函数调用,还是某个 URL 点击,客户端能够根据用户的点击或调用的函数,确定该插件的标识,例如 AppID。

[0048] 步骤 202、应用程序客户端根据所述插件标识信息,获取与该插件标识信息对应的配置文件。

[0049] 该步骤中,若配置文件保存在客户端本地,则从客户端本地获取与该插件标识信息对应的配置文件;若配置文件保存在指定服务器中,则从指定服务器获取与该插件标识信息对应的配置文件。

[0050] 获取配置文件时,根据插件标识信息查找对应的配置文件的存储路径,并根据查找到的存储路径获取对应的配置文件。

[0051] 步骤 203、根据获取的配置文件判断该插件是否需要注册,若判断是,则执行步骤 204,否则执行步骤 206。

[0052] 该步骤中,获取的配置文件包括插件是否需要注册的描述信息,对应一些简单应用的插件,则无需注册。

[0053] 步骤 204、根据获取的配置文件包括的注册信息将该插件注册到客户端。

[0054] 该步骤中,配置文件包括的注册信息包括如下注册项中的一个或多个:

[0055] 注册表信息、环境变量信息、公共运行库信息以及插件自定义安装文件信息。

[0056] 具体地,将用户请求使用的插件注册到系统时,首先确定该注册信息中指定的注册项;若注册项为一个,则执行该注册项对应的注册过程;若注册项为多个,则依次执行各注册项分别对应的注册过程。

[0057] 步骤 205、判断插件注册是否成功,若是,则执行步骤 206,否则结束。

[0058] 该步骤中,注册信息中包括的注册项为多个时,每个注册项分别对应的注册过程都执行成功,则确定插件注册成功,否则确定插件注册不成功。

[0059] 步骤 206、接受用户的插件使用请求,允许用户使用其请求使用的插件。

[0060] 以下进一步以注册信息包括的注册项为注册表信息、环境变量信息、公共运行库信息以及插件自定义安装文件信息为例,对各注册项对应的注册过程进行具体描述:

[0061] 1、注册项为注册表信息时,对应的注册过程包括:

[0062] 将注册表信息直接写入系统注册表中,若写入失败,则系统返回写入失败信息,此次注册失败。

[0063] 2、注册项为环境变量信息时,对应的注册过程包括:

[0064] 根据环境变量信息执行设定的环境变量,若执行失败,则系统返回执行失败信息,此次注册失败。

[0065] 3、注册项为公共运行库信息时,对应的注册过程包括:

[0066] 首先提示用户需要安装运行库文件,如果用户拒绝则注册失败,否则根据插件所支持的运行库环境从服务器下载指定运行库安装文件(例如,下载的运行库文件可以由微软或者 SUN 提供标准安装包),然后将文件安装到指定目录下,安装完成后检查指定目录下是否有相应的安装文件包,若有,则安装成功,否则失败。

[0067] 公共运行库信息为插件支持的运行库环境信息,并可以进一步包括公共运行库信息的下载地址,若公共运行库信息中不包括公共运行库信息的下载地址,则客户端根据默认的下载地址下载对应的公共运行库文件。

[0068] 插件所支持的运行库环境可以为 VB 运行库、VC 运行库、Net Framework 运行库、Java 运行库或 DirectX 运行库等,具体可以根据需要扩展。

[0069] 4、注册项为插件自定义安装文件信息时,对应的注册过程包括:

[0070] 插件自定义安装文件一般是由插件开发者开发并且部署到用户客户端中,具体地,该插件自定义安装文件信息可以为该插件自定义安装文件的下载地址,客户端可以根据该下载地址预先下载或在注册时实时下载插件自定义安装文件。

[0071] 具体执行注册时,对预先下载或实时下载的插件自定义安装文件执行复制操作,即将插件自定义安装文件复制到指定目录下,例如,复制到系统目录下。然后检查对应的目录下是否有该插件自定义安装文件,若有,则注册成功,否则失败。

[0072] 进一步地,上述插件自定义安装文件还可以由插件自带,较佳地,如果插件自定义安装文件由插件自带,则无需在配置文件的注册信息中设置插件自定义安装文件信息。

[0073] 需要指出的是,上述实施例中的注册信息中的注册项并不限于以上四种,可以根据需要扩展。

[0074] 又一实施例中,上述步骤 202 获取到的配置文件还包括插件对应的运行库信息,上述流程还应包括步骤:

[0075] 应用程序客户端根据该运行库信息,获取对应的运行库安装文件并安装。

[0076] 该步骤具体在上述流程的步骤 202 之后以及步骤 205 之前执行,并且,在步骤 205 中,判断插件注册成功,具体为:

[0077] 注册信息中包括的注册项为多个时,每个注册项分别对应的注册过程都执行成功,以及,根据该运行库信息成功获取对应的运行库安装文件并成功安装。

[0078] 与上述流程对应,本申请实施例还提供了一种客户端,如图 3 所示,该客户端,包括:

[0079] 接收单元 301、配置文件获取单元 302 以及注册单元 303。

[0080] 较佳地,配置文件获取单元 302 包括:第一获取模块 3021,或,第二获取模块 3022。

[0081] 较佳地,注册单元 303 包括:注册项确定模块 3031,和,注册执行模块 3032。

[0082] 其中:

[0083] 接收单元 301,用于接收插件使用请求并根据接收的插件使用请求获取插件标识信息;

[0084] 配置文件获取单元 302,用于根据接收单元 301 获取的插件标识信息,获取与该插件标识信息对应的配置文件;

[0085] 注册单元 303,用于根据配置文件获取单元 302 获取的配置文件包括的注册信息将请求使用的插件进行注册。

[0086] 一个实施例中,配置文件获取单元 302 包括的第一获取模块 3021,用于从客户端预先下载到本地的配置文件中获取与插件标识信息对应的配置文件;第二获取模块 3022,用于从服务器获取与插件标识信息对应的配置文件。

[0087] 一个实施例中,注册单元 303 包括的注册项确定模块 3031,用于确定配置文件获取单元 302 获取的配置文件包括的注册信息中指定的注册项;注册执行模块 3032,用于当注册项确定模块 3031 确定的注册项为一个时,执行该注册项对应的注册过程;当注册项确定模块 3031 确定的注册项为多个时,依次执行各注册项分别对应的注册过程。

[0088] 一个实施例中,配置文件获取单元 302 获取的配置文件还包括:插件对应的运行库信息;

[0089] 相应地,注册单元 303 还用于,根据该运行库信息,获取对应的运行库安装文件并安装。

[0090] 通过以上技术方案,本申请实施例中,应用程序客户端接收插件使用请求后,获取相应的插件标识信息,进一步获取对应的配置文件,并根据该配置文件包括的注册信息将请求使用的插件进行注册。采用该技术方案,由应用程序客户端根据预设的配置文件执行插件的注册过程,因此,插件无需自带安装包,从而省去了插件开发者对安装包的制作过程,减轻了插件开发者以及用户的负担,并且采用该技术方案,插件除了适用于对插件是否注册无特殊要求的应用程序,也适用于要求插件必须注册的应用程序,从而提高了插件的适用范围。

[0091] 为了描述的方便,以上所述客户端的各部分以功能分为各种模块或单元分别描述。当然,在实施本申请时可以把各模块或单元的功能在同一个或多个软件或硬件中实现。

[0092] 显然,本领域的技术人员可以对本申请进行各种改动和变型而不脱离本申请的精神和范围。这样,倘若本申请的这些修改和变型属于本申请权利要求及其等同技术的范围

之内,则本申请也意图包含这些改动和变型在内。

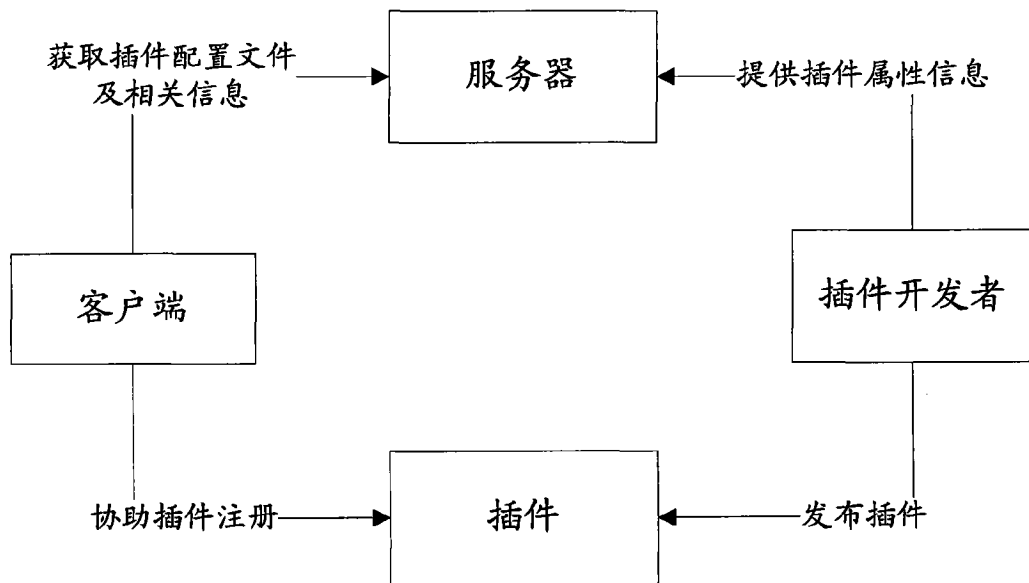


图 1

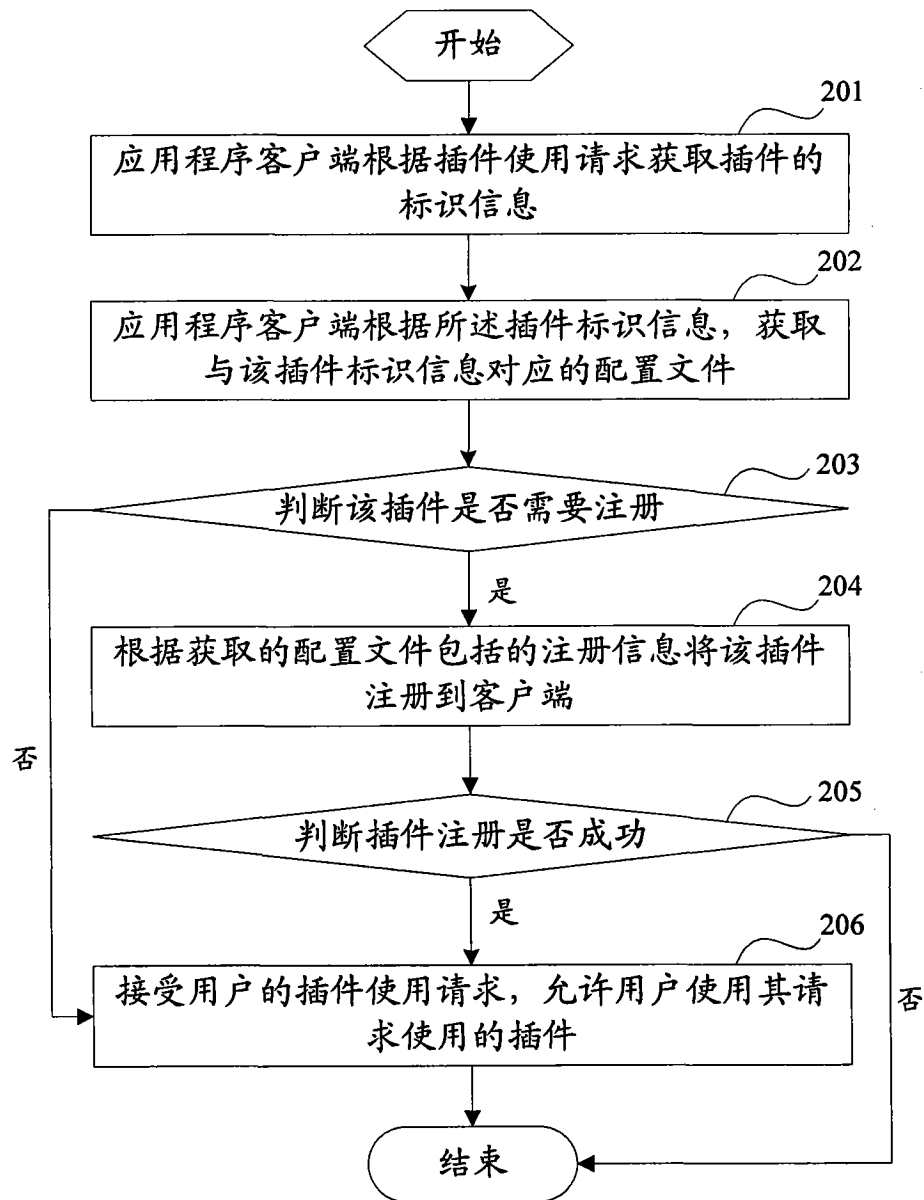


图 2

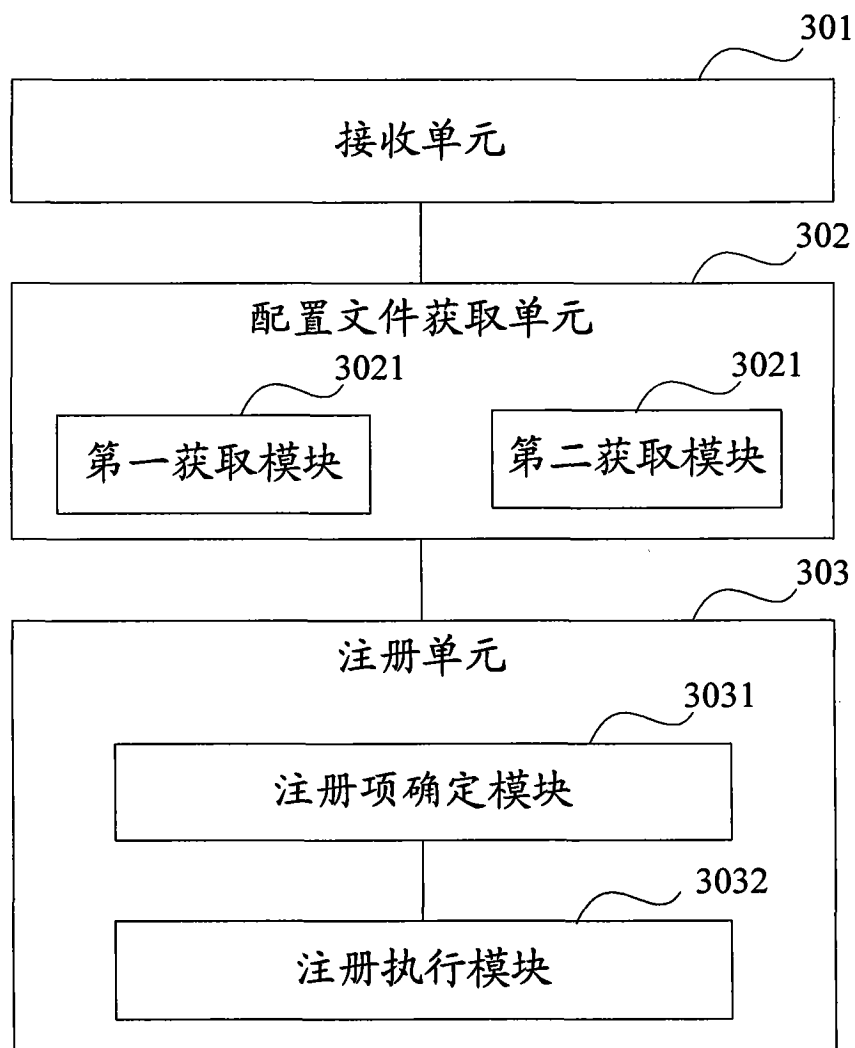


图 3