



(12) 发明专利

(10) 授权公告号 CN 1866214 B

(45) 授权公告日 2010.12.08

(21) 申请号 200610078389.7

审查员 李乐

(22) 申请日 2006.05.17

(30) 优先权数据

2005-144225 2005.05.17 JP

(73) 专利权人 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 牛久丰彦

(74) 专利代理机构 北京林达刘知识产权代理事

务所(普通合伙) 11277

代理人 刘新宇 权鲜枝

(51) Int. Cl.

G06F 9/445(2006.01)

(56) 对比文件

US 20030172175 A1, 2003.09.11, 说明书第 52 段, 第 53 段, 第 65 段, 第 66 段至第 71 段、图 1-4.

US 20020138666 A1, 2002.09.26, 摘要, 说明书第 54 段, 第 57 段, 第 61 段, 图 12.

JP 2004062892 A, 2004.02.26, 全文.

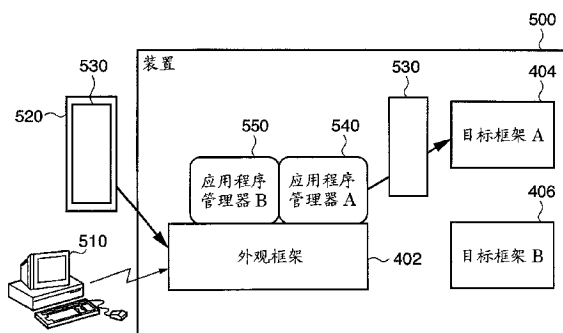
权利要求书 2 页 说明书 11 页 附图 14 页

(54) 发明名称

安装方法及设备

(57) 摘要

本发明提供一种安装方法及设备。在安装应用程序时, 多个框架中的一个框架判断该应用程序要被安装到的框架, 并将该应用程序安装在所判断出的框架中。



1. 一种由管理框架执行的管理方法,所述管理方法包括:
接收步骤,用于接收来自操作者的指令;
判断步骤,用于根据关于安装应用程序的安装指令并基于附加至该应用程序的信息,判断该应用程序要被安装到的、包括所述管理框架的多个框架中的一个框架;
安装步骤,用于将所述应用程序安装在所述判断步骤中判断出的所述多个框架中的所述一个框架中;
启动步骤,用于根据启动指令,启动安装在所述多个框架中的所述一个框架中的所述应用程序;以及
停止步骤,用于根据停止指令,停止所述应用程序。
2. 根据权利要求1所述的管理方法,其特征在于,在所述安装步骤中,所述应用程序被安装在所述管理框架中。
3. 根据权利要求1所述的管理方法,其特征在于,在所述安装步骤中,所述应用程序被安装在所述管理框架以外的、所述多个框架中的所述一个框架中。
4. 根据权利要求1所述的管理方法,其特征在于,所述多个框架由操作系统作为运行于该操作系统上的应用程序来处理。
5. 根据权利要求1所述的管理方法,其特征在于,当所述管理框架以外的、所述多个框架中的所述一个框架上的应用程序被选择时,向已经安装所选择的所述应用程序的、所述管理框架以外的、所述多个框架中的所述一个框架请求处理。
6. 根据权利要求1所述的管理方法,其特征在于,由所述管理框架来管理所述管理框架以外的、所述多个框架中的所述一个框架上的应用程序。
7. 根据权利要求1所述的管理方法,其特征在于,在判断为要在所述管理框架以外的、所述多个框架中的所述一个框架中安装所述应用程序的情况下,生成对应于要安装的所述应用程序的应用程序,以在所述管理框架上管理要安装的所述应用程序。
8. 根据权利要求1所述的管理方法,其特征在于,管理所述多个框架上的分组的应用程序。
9. 根据权利要求1所述的管理方法,其特征在于,所述安装步骤通过所述管理框架调用处理所述管理框架上的应用程序的应用程序管理器的安装方法。
10. 根据权利要求1所述的管理方法,其特征在于,所述安装步骤使用由所述管理框架提供的搜索方法,搜索用于将所述应用程序安装在所述多个框架中的所述一个框架中的应用程序管理器。
11. 一种包括管理框架的信息处理设备,其中,所述管理框架包括:
接收装置,用于接收来自操作者的指令;
判断装置,用于根据关于安装应用程序的安装指令并基于附加至该应用程序的信息,判断该应用程序要被安装到的、包括所述管理框架的多个框架中的一个框架;
安装装置,用于将所述应用程序安装在由所述判断装置判断出的所述多个框架中的所述一个框架中;
启动装置,用于根据启动指令,启动安装在所述多个框架中的所述一个框架中的所述应用程序;以及
停止装置,用于根据停止指令,停止所述应用程序。

12. 根据权利要求 11 所述的信息处理设备,其特征在于,所述安装装置将所述应用程序安装在所述管理框架中。

13. 根据权利要求 11 所述的信息处理设备,其特征在于,所述安装装置将所述应用程序安装在所述管理框架以外的、所述多个框架中的所述一个框架中。

14. 根据权利要求 11 所述的信息处理设备,其特征在于,所述多个框架由操作系统作为运行于该操作系统上的应用程序来处理。

15. 根据权利要求 11 所述的信息处理设备,其特征在于,当所述管理框架以外的、所述多个框架中的所述一个框架上的应用程序被选择时,向已经安装所选择的所述应用程序的、所述管理框架以外的、所述多个框架中的所述一个框架请求处理。

16. 根据权利要求 11 所述的信息处理设备,其特征在于,所述管理框架还包括管理装置,该管理装置用于管理所述管理框架以外的、所述多个框架中的所述一个框架上的应用程序。

17. 根据权利要求 11 所述的信息处理设备,其特征在于,所述管理框架还包括生成装置,该生成装置用于在判断为要在所述管理框架以外的、所述多个框架中的所述一个框架中安装所述应用程序的情况下,生成与要安装的所述应用程序相对应的应用程序,以管理要安装的所述应用程序。

18. 根据权利要求 11 所述的信息处理设备,其特征在于,所述管理框架还包括管理装置,该管理装置用于管理所述多个框架上的分组的应用程序。

19. 根据权利要求 11 所述的信息处理设备,其特征在于,所述安装装置通过所述管理框架调用处理所述管理框架上的应用程序的应用程序管理器的安装方法。

20. 根据权利要求 11 所述的信息处理设备,其特征在于,所述安装装置使用由所述管理框架提供的搜索方法,搜索用于将所述应用程序安装在所述多个框架中的所述一个框架中的应用程序管理器。

安装方法及设备

技术领域

[0001] 本发明涉及一种用于安装应用程序的方法及设备。

背景技术

[0002] 传统上,有被称作框架 (framework)、平台、应用程序服务器、程序容器 (program container) 等的多种软件,它们被用于进行例如计算机系统的服务器上的应用程序的执行控制。以下将这种软件通称为框架。

[0003] 该框架具有所谓的“应用程序生命周期管理功能”。该“应用程序生命周期管理功能”实现下列执行控制。

[0004] - 在框架上安装要被执行的应用程序

[0005] - 卸载不需要的应用程序

[0006] - 执行所安装的应用程序

[0007] - 停止正在运行的应用程序

[0008] - 更新所安装的应用程序等

[0009] 在各框架上安装的应用程序的管理方法和形式依框架而不同。

[0010] 作为典型的框架,“The Apache Jakarta Tomcat 5.5Servlet/JSP Container UserGuide”(Apache Project,2004)和“OSGi Service Platform Specification Release 2”(OSGI,2001)是公知的。

[0011] 在“The Apache Jakarta Tomcat 5.5 Servlet/JSP ContainerUserGuide”中,被称作 WAR 格式的应用程序分发形式是普遍的。如果该 WAR 文件被存储在特定的文件夹中,则识别出新存储了 WAR 文件,并将该文件自动转换为可执行的形式,由此执行安装。另外,还支持使用该机理管理应用程序生命周期的应用程序,并且可从远程对应用程序进行安装、卸载、启动、停止等。

[0012] 在“OSGi Service Platform Specification Release 2”中,定义被称作“包 (bundle)”的应用程序的单位,并且还定义形成该包的包文件的形式。对这些包提供生命周期管理功能。

[0013] 另一方面,在通用计算机以外的所谓“嵌入式装置”的系统中,随着装置的硬件资源的速度和容量的增加,在许多情况下,由通用计算机开发的技术将被用于嵌入式装置。

[0014] 在通用计算机系统的服务器中,管理服务器的硬件和软件的操作者已经接受过专业培训。因此,即使当应用程序的管理方法依框架而不同时,如果假定他们具有处理这种差别的技能,则不会造成问题。

[0015] 另一方面,在嵌入式装置中,在多数情况下,不期望具有高专业技能的人被指派为实际管理该装置的操作者。

[0016] 因此,当应用程序的管理方法对各个框架不同时,很可能因缺乏经验和知识而发生操作错误。也就是说,整个系统很可能发生功能障碍。此外,甚至管理服务器的操作者也很难说不出现操作错误。

[0017] 此外,通常大多在安装时进行应用程序的许可管理、安全检查等。因此,如果作为应用程序的生命周期管理方法的一部分的安装方法对各个框架不同,则在该系统中必须准备多个许可管理功能和安全检查功能。也就是说,硬件资源的使用效率非期望地降低。

[0018] 此外,用于远程控制的网络进入路径必须对各个框架的外部开放,并且有许多允许从外部侵入该系统的途径(route)。因此,这种途径可成为安全方面的瓶颈。

发明内容

[0019] 本发明是为解决上述问题而做出的。

[0020] 本发明的目的是防止操作者的操作错误。

[0021] 本发明的另一目的是防御外部的侵入。

[0022] 本发明的另一目的是提供一种由管理框架执行的管理方法,包括:接收步骤,用于接收来自操作者的指令;判断步骤,用于根据关于安装应用程序的安装指令并基于附加至该应用程序的信息,判断该应用程序要被安装到的、包括所述管理框架的多个框架中的一个框架;安装步骤,用于将所述应用程序安装在所述判断步骤中判断出的所述多个框架中的所述一个框架中;启动步骤,用于根据启动指令,启动安装在所述多个框架中的所述一个框架中的所述应用程序;以及停止步骤,用于根据停止指令,停止所述应用程序。

[0023] 本发明的另一目的是提供一种包括管理框架的信息处理设备,其中,所述管理框架包括:接收装置,用于接收来自操作者的指令;判断装置,用于根据关于安装应用程序的安装指令并基于附加至该应用程序的信息,判断该应用程序要被安装到的、包括所述管理框架的多个框架中的一个框架;安装装置,用于将所述应用程序安装在由所述判断装置判断出的所述多个框架中的所述一个框架中;启动装置,用于根据启动指令,启动安装在所述多个框架中的所述一个框架中的所述应用程序;以及停止装置,用于根据停止指令,停止所述应用程序。

[0024] 通过下面结合附图的说明,本发明的其它特征和优点将是很明显的,其中,在其全部附图中,同样的附图标记表示相同或相似的部分。

[0025] 附图说明

[0026] 图1是示出一般的嵌入式装置的结构例子;

[0027] 图2示出通过Web浏览器显示在计算机上的管理窗口;

[0028] 图3是示出第一实施例中的嵌入式装置的结构例子的框图;

[0029] 图4是分层地示出由控制单元301执行的软件的配置的视图;

[0030] 图5是说明外观框架(facade framework)402和目标框架A404之间的关系的图;

[0031] 图6示出第一实施例中应用程序文件520的结构例子;

[0032] 图7示出第一实施例中程序属性的说明例子;

[0033] 图8是示出第一实施例中安装处理的流程图;

[0034] 图9示出第一实施例中界面的例子;

[0035] 图10是示出第一实施例中应用程序的停止处理的流程图;

[0036] 图11示出由外观框架402管理的应用程序组403的应用程序属性对象的配置;

[0037] 图12示出第三实施例中使用的应用程序文件的格式;

[0038] 图13示出第三实施例中程序属性的说明例子;

- [0039] 图 14 是示出第三实施例中安装处理的流程图；
- [0040] 图 15 示出第三实施例中 ApplicationAttribute 对象的例子；以及
- [0041] 图 16 是示出第三实施例中应用程序的停止处理的流程图。
- [0042] 具体实施方式
- [0043] 下面将参照附图对本发明的优选实施例进行详细说明。
- [0044] 下面将使用图 1 说明嵌入有多个框架的嵌入式装置的结构和操作。注意，本发明不局限于该嵌入式装置。
- [0045] 图 1 是示出一般的嵌入式装置的结构例子。参看图 1，附图标记 100 表示运行框架的嵌入式装置（以下简称为“装置”）。附图标记 101 表示框架 A，作为运行在装置 100 上的框架。附图标记 102 表示应用程序组 A，作为在框架 A 101 上被管理的应用程序。附图标记 110 表示指示框架 A 101 管理应用程序的计算机 A。附图标记 120 表示将经由计算机 A 110 安装到框架 A 101 中的应用程序 A。
- [0046] 附图标记 103 表示框架 B，作为运行在装置 100 上的框架。附图标记 104 表示应用程序组 B，作为在框架 B 103 上被管理的应用程序。附图标记 130 表示指示框架 B 103 管理应用程序的计算机 B。附图标记 140 表示将经由计算机 B 130 安装到框架 B 103 中的应用程序 B。
- [0047] 可使用 WWW(World Wide Web, 万维网) 技术从远程（计算机 A 110 或者计算机 B 130）控制这些应用程序管理功能。也就是说，使用在不同于实际启动框架（A 101 或 B 103）的装置 100 的计算机上运行的 Web 浏览器，从远程访问装置 100 中的框架（A 101 或 B 103）。框架（A 101 或 B 103）上的应用程序（A 102 或 B 104）的生命周期可被管理。
- [0048] 图 2 示出通过 Web 浏览器在计算机上显示的管理窗口的例子。该例子是当计算机 A 110 管理框架 A 101 上的应用程序组 A 102 时，通过运行在计算机 A 110 上的 Web 浏览器显示的应用程序管理窗口。
- [0049] 计算机 A 110 的用户从该管理窗口使用选择按钮 201 来选择存储应用程序的文件。在用户使用选择按钮 202 指定了存储许可信息的文件之后，他或她使用安装按钮 203 开始安装。
- [0050] 此外，用户从该管理窗口选择已安装的应用程序（例如，应用程序 A-2），并指示执行控制，例如启动 204、停止 205、更新 206、卸载 207 等。
- [0051] 另一方面，框架 B 103 通过使用 FTP(File Transfer Protocol, 文件传送协议) 将应用程序 B 140 传送到由框架 B 103 确定的特定文件夹来安装应用程序。与安装相同，通过使用 FTP 从特定的文件夹中删除文件来完成卸载。
- [0052] 这样，甚至同一装置 100 上的框架 A 101 和 B 103 也具有不同的应用程序管理方法。
- [0053] 注意，为了说明方便，将计算机 A 110 和 B 130 说明为不同的计算机，但是它们可以是一个计算机。
- [0054] 第一实施例
- [0055] 下面将说明统一管理运行有多个框架的装置上的应用程序的生命周期的方法。
- [0056] 图 3 是示出第一实施例中嵌入式装置的结构例子的框图。注意，本发明不局限于该嵌入式装置，并且可将本发明应用于其它网络装置。

[0057] 参看图 3, 附图标记 301 表示控制整个装置的控制单元。附图标记 302 表示一次存储单元, 包括 RAM 等, 用于临时存储数据和将由控制单元 301 执行的软件。附图标记 303 表示二次存储单元, 包括硬盘等, 用于永久存储数据和将由控制单元 301 执行的软件。附图标记 304 表示输入单元, 包括键盘、鼠标、触摸屏、扫描器等, 用于输入用户指令、文本数据和图像数据。附图标记 305 表示显示单元, 包括显示器, 用于显示对操作者的指示和数据内容。附图标记 306 表示网络单元, 包括 LAN 等, 用于与其它装置通信。附图标记 307 表示相互连接各个单元的总线。

[0058] 图 4 分层示出将由控制单元 301 执行的软件的配置。参看图 4, 附图标记 401 表示用于管理软件和硬件资源的操作系统 (OS)。附图标记 402 表示接收操作者指令的外观框架, 作为用作应用程序执行环境的框架的代表。附图标记 403 表示运行在外观框架 402 上的外观框架应用程序组。

[0059] 附图标记 404 表示目标框架 A, 作为经由外观框架 402 接收操作者指令的框架。附图标记 405 表示运行在目标框架 A 404 上的目标框架 A 应用程序组。附图标记 406 表示与目标框架 A 404 相同, 经由外观框架 402 接收操作者指令的目标框架 B。附图标记 407 表示运行在目标框架 B 406 上的目标框架 B 应用程序组。附图标记 408 表示运行在 OS 401 上的其它应用程序组。

[0060] 注意, 与属于其它应用程序组 408 的应用程序同等地, 从 OS401 对外观框架 402、目标框架 A 404 和目标框架 B 406 进行处理。

[0061] 在第一实施例中, 将外观框架 402 作为 OSGi ServicePlatform (OSGi 服务平台, 以下简称为 OSGi) 的扩展的外观框架进行说明。

[0062] 外观框架 402 由 Java 来构建 (Java 是美国 Sun Microsystems 公司的注册商标)。每个应用程序被称作包, 并且作为档案文件 (archive file) 格式的 Jar (Java ARchive) 被提供。外观框架 402 具有将所给的应用程序的一些功能提供给另一应用程序的服务登记、搜索和使用功能。利用这些功能, 具有特定接口的服务可被搜索和使用。在服务登记功能中, 指定要被登记的服务的属性可被一起登记, 并且可基于该属性搜索和获得具有特定属性值的服务。注意, OSGi 的细节在上述 “OSGi Service Platform Specification Release 2” 中被说明, 并将省略其说明。

[0063] 图 5 说明了外观框架 402 和目标框架 404 之间的关系。参看图 5, 附图标记 500 表示第一实施例中的装置, 其包括图 3 中所示的各个单元。附图标记 510 表示经由网络单元 306 连接到装置 500 的计算机。附图标记 520 表示存储在计算机 510 的存储装置中并包括将经由外观框架 402 安装在目标框架 A 404 中的目标应用程序的应用程序文件。附图标记 530 表示包括在应用程序文件 520 中并运行在目标框架 A 404 上的目标应用程序。

[0064] 即使当在不同的装置中分别准备外观框架 402 和目标框架 404 及 406 时, 也可应用本发明。在此情况下, 在给定的装置中准备的外观框架 402 将目标应用程序安装到在其它装置中准备的目标框架 404 或 406 中。

[0065] 下面将使用图 6 说明第一实施例中的应用程序文件 520 的结构。

[0066] 图 6 示出第一实施例中应用程序文件 520 的结构例子。如图 6 所示, 应用程序文件 520 具有头区域 601 和数据存储区域 602。

[0067] 头区域 601 包括表示应用程序文件的幻数 (magic number) 611、表示头区域 601 的

大小的头区域大小 612、以及说明该应用程序的各种属性的程序属性说明区域 613。注意，如图 7 所示，程序属性说明区域 613 具有表示程序属性的 Jar 文件的清单头（manifest header）格式。注意，该清单头说明作为一对给定属性的名称和值的属性值。

[0068] 另一方面，数据存储区域 602 被加密，使得只由外观框架 402 管理的密钥来解密。当数据存储区域 602 被解密时，它被分成目标框架应用程序头区域 621 和程序数据存档区域 622。目标框架应用程序头区域 621 说明在目标框架 A 404 上执行时所需的属性。程序数据存档区域 622 存储将在目标框架上执行的程序数据。

[0069] 参看图 5，附图标记 540 表示应用程序管理器 A，其属于外观框架应用程序组 403，在外观框架 402 上执行，并相对于目标框架 A 404 进行应用程序的生命周期管理。附图标记 550 表示应用程序管理器 B，其属于外观框架应用程序组 403，在外观框架 402 上执行，并相对于目标框架 B 406 进行应用程序的生命周期管理。也就是说，与框架相一致地存在进行应用程序的生命周期管理的应用程序管理器。

[0070] 下面将使用图 8 说明当计算机 510 的操作者将应用程序文件 520 安装在目标框架 A 404 中时所执行的处理。假定操作者同时指定许可信息和应用程序文件 520。此外，使用与图 2 中所示的相同的管理窗口作为在第一实施例中使用的 Web 浏览器上的用户界面。

[0071] 图 8 是示出第一实施例中的安装处理的流程图。在步骤 S801，计算机 510 的操作者经由装置 500 中的外观框架 402 的管理用户界面访问外观框架 402，该管理用户界面通过 Web 浏览器来显示。然后操作者指示安装应用程序文件 520。

[0072] 当响应于操作者的指令接收到应用程序文件 520 时，外观框架 402 解释该应用程序文件 520 的头区域 601 中的程序属性说明区域 613 中所说明的清单头。然后，外观框架 402 从该清单头中获得将执行存储在程序数据存档区域 622 中的程序的框架。

[0073] 注意，如图 7 所示，该用作目标的框架被说明为目标（target：）属性。当没有设定目标属性时，或者当指定了外观框架 402 时，外观框架 402 被识别为目标框架。在此情况下，“Tomcat 5.5，，被指定为目标，并且目标框架 A 404 被识别出。

[0074] 在步骤 S802 中检查在步骤 S801 中获得的并被用于执行应用程序文件 520 的目标框架是否为外观框架 402。如果该目标框架是外观框架 402，则流程前进到步骤 S803。

[0075] 在步骤 S803，由操作者指定的并从计算机 510 发送到外观框架 402 的应用程序文件 520 被安装作为外观框架 402 的应用程序。

[0076] 当安装时，使用密钥来解密存储在应用程序文件 520 的数据存储区域 602 中的数据。解密后的数据被假定是外观框架 402 的应用程序，并在外观框架 402 中进行安装处理。

[0077] 此外，通过外观框架 402 的功能来实现用于检查解密后的数据是否符合外观框架 402 的应用程序的格式的处理。此外，由操作者指定的许可信息被设定为外观框架 402 的许可信息。

[0078] 另一方面，如果在步骤 S802 中识别出外观框架 402 以外的框架是目标框架，则流程前进到步骤 S804 以进行搜索，检查是否存在用于所识别出的目标框架的应用程序管理器。注意，可在不同的装置中准备外观框架 402 和目标框架。

[0079] 应用程序管理器被登记作为使用图 9 所示的接口（ApplicationManager）来提供服务的应用程序管理器。作为该应用程序管理器的属性，作为目标的框架的标识符被作为“TargetFramework”属性登记。

[0080] 在该例子中,在步骤 S801 中识别出目标框架是“Tomcat 5.5”。因此,当在外观框架 402 主体所提供的搜索方法中以 LDAP 格式指定 ApplicationManager 接口的名称以及其属性值的搜索公式如下时,可获得用于“Tomcat 5.5”的应用程序管理器。

[0081] `ApplicationManager am = (ApplicationManager)framework_body.
search(“com.canon.upd.ApplicationManager”,“(TargetFramework = Tomcat 5.5)”);`

[0082] 当没有用于“Tomcat 5.5”的应用程序管理器被登记时,返回空 (Null),以识别该事实。

[0083] 也就是说,如果目标框架 A 404 是“Tomcat 5.5”,则可通过上述搜索获得目标框架 A 404 的应用程序管理器。

[0084] 如果成功得到目标框架的应用程序管理器,则流程前进到步骤 S806;如果不能获得应用程序管理器并且返回空,则流程前进到步骤 S805。

[0085] 在步骤 S805,将表示没有与指定为该应用程序的安装目的地的目标框架相对应的应用程序管理器被登记的警告页显示在计算机 510 的 Web 浏览器上。该警告页通知操作者该安装已经失败。

[0086] 如果在步骤 S804 中可获得与指定为该应用程序的安装目的地的目标框架相对应的应用程序管理器,则流程前进到步骤 S806。在此情况下,作为目标框架 A 404 的应用程序管理器的应用程序管理器 A 540 被返回。

[0087] 如果应用程序管理器存在,则流程前进到步骤 S806,以从应用程序文件 520 中取出数据存储区域 602 并使用密钥对其进行解密。解密后的目标应用程序 530 作为输入流被打开,并以该流作为变量调用应用程序管理器 A 540 的安装方法,以将该应用程序安装在目标框架 A 404 中。

[0088] 在此情况下,由于目标框架 A 404 是 Tomcat,所以由该输入流传递的数据作为 WAR 文件存储在特定文件夹中,由此执行安装处理。

[0089] 当完成在目标框架 A 404 中的安装时,实现图 9 所示 Application 接口的对象被返回。实现该 Application 接口的对象包括用于唯一识别目标框架 A 404 上的目标框架 A 应用程序组 405 中所安装的应用程序的应用程序标识符。

[0090] 可通过调用该 Application 接口的 `getApplicationID` 方法来获得该应用程序标识符。

[0091] 目标框架 A 404 被告知由应用程序管理器 A 540 指定的许可信息,并且由目标框架 A 404 来使用该许可信息。

[0092] 当使用所获得的 Application 对象和关于外观框架 402 的主体的目标框架的标识符(在该例子中为字符串“Tomcat 5.5”)调用作为扩展方法的 `makeBundle` 方法时,虚拟应用程序程序(virtualapplication)被产生。该虚拟应用程序是在外观框架 402 上产生的虚拟应用程序,并对应于目标框架 A 404 上所安装的应用程序。

[0093] 对该虚拟应用程序进行目标框架 A 404 中安装的应用程序的随后的生命周期管理。在此情况下,执行控制可被进行,好像外观框架 402 上的应用程序被管理。

[0094] 如上所述,该应用程序可被安装在目标框架 A 404 中,从操作者的角度看,好像它被安装在外观框架 402 中。

[0095] 下面将说明安装以外的处理。安装以外的各处理大致相同,将使用图 10 来说明作

为代表性例子的用于停止运行中的应用程序的处理。

[0096] 图 10 是示出第一实施例中应用程序的停止处理的流程图。与上述安装处理中相同,操作者从计算机 510 访问外观框架 402,以将外观框架 402 的应用程序组 403 的管理窗口显示在 Web 浏览器上。当操作者从所显示的管理窗口选择应用程序并指示进行停止处理时,流程前进到步骤 S1001。应用程序组 403 包括外观框架 402 本身的应用程序和目标框架上的虚拟应用程序。在步骤 S1001 中识别外观框架应用程序组 403 中被选择的应用程序是否为目标框架上的应用程序的虚拟应用程序。基于图 11 所示的应用程序属性对象来完成该识别处理。注意,可在不同的装置中准备外观框架 402 和目标框架。

[0097] 图 11 示出由外观框架 402 管理的应用程序组 403 的应用程序属性对象的配置。

[0098] 如图 11 所示,应用程序属性对象包括下列信息:

- [0099] - 框架标识符 1101,表示执行应用程序的实体的框架;
- [0100] - 应用程序标识符 1102,表示被定义为 Application 接口的对象的应用程序;
- [0101] - 状态 1103,表示应用程序的生命周期状态;
- [0102] - 存储位置 1104,表示作为 URL 信息的应用程序被实际存储的位置;以及
- [0103] - 其它属性 1105。

[0104] 当该应用程序被安装时(步骤 S806),从所获得的 Application 对象获得并设置目标框架上的该应用程序的属性值。

[0105] 如果框架标识符 1101 被设定,则表示所指定的应用程序是有所设定的标识符的目标框架上的应用程序。如果框架标识符 1101 没有被设定,则表示所指定的应用程序是外观框架 402 上的应用程序。

[0106] 如果在步骤 S1001 中确定为所选择的应用程序是外观框架 402 上的应用程序,则流程前进到步骤 S1002,否则,流程前进到步骤 S1003。

[0107] 在步骤 S1002,由于所选择的应用程序是正实际运行在外观框架 402 上的应用程序,因此外观框架 402 的原始应用程序管理中的应用程序停止处理被执行。

[0108] 在步骤 S1003,与由该应用程序属性对象的框架标识符 1101 选择的并且该应用程序正实际运行于其上的目标框架相对应的应用程序管理器被获取。该步骤中的获取方法与步骤 S804 中所说明的获取方法相同。

[0109] 然后在步骤 S1004 中检查该应用程序管理器是否可被获得。如果该应用程序管理器可被获得,则流程前进到步骤 S1005;否则,生成异常,从而结束该处理。

[0110] 在步骤 S1005,正运行于该目标框架上的所指定的应用程序被停止。更具体来说,将存储在应用程序标识符 1102 中的该应用程序标识符指定为关于在步骤 S1003 中获得的程序管理器的变量,以调用停止方法。

[0111] 同时,流程前进到步骤 S1002,以停止外观框架 402 上的虚拟应用程序。

[0112] 如上所述,与目标框架上的应用程序相关联的生命周期管理可被进行,从操作者的角度看,好像外观框架 402 上的应用程序的生命周期管理被进行。

[0113] 当外观框架 402 上的应用程序的应用程序属性对象被生成时,同时将其存储在二次存储单元 303 中。之后,当装置 500 被启动时,从二次存储单元 303 加载该应用程序属性对象并将其恢复。

[0114] 因此,当经由外观框架 402 安装应用程序时,其信息被恢复到每次启动装置 500 之

前装置 500 被关闭的状况。

[0115] 此时,在外观框架的启动序列中,当对外观框架 402 上的虚拟应用程序执行恢复处理时,进行下列处理。

[0116] 当存储在该虚拟应用程序的应用程序属性对象中的应用程序的状态不同于每个目标框架上的应用程序的状态时,目标框架上的应用程序的状态被改变。也就是说,外观框架 402 通过对应的应用程序管理器改变目标框架上的应用程序的状态,以匹配外观框架 402 上的虚拟应用程序的状态。

[0117] 例如,尽管表示启动状况的状态被存储在外观框架 402 上,但是如果在目标框架上没有启动对应的应用程序,则通过该应用程序管理器的启动方法将该状态改变为启动状态。

[0118] 第二实施例

[0119] 下面将参照附图详细说明根据本发明的第二实施例。注意,第二实施例中的装置和软件的结构与第一实施例中的相同,并将省略其说明。此外,可以在不同的装置中准备外观框架和目标框架。

[0120] 第一实施例与第二实施例的差别在于当装置 500 被启动时外观框架 402 和应用程序管理器的处理。

[0121] 当装置 500 被启动,并且外观框架 402 也被相应启动时,外观框架应用程序组 403 被初始化。在此情况下,当作为虚拟应用程序的应用程序属性对象的框架标识符被设定时,在不重构该应用程序的情况下,该虚拟应用程序不被再现为外观框架 402 上的虚拟应用程序。

[0122] 代替此时禁止再现,在虚拟应用程序的情况下,该应用程序属性对象不被存储在二次存储单元 303 中,从而获得相同的效果。

[0123] 另一方面,当通过外观框架 402 启动应用程序管理器时,正运行在每个目标框架上的应用程序的应用程序标识符列表被获得。通过向将由该应用程序管理器管理的目标框架发送询问来获得该应用程序标识符列表。

[0124] 通过使用所获得的列表上的应用程序标识符,在外观框架 402 上生成虚拟应用程序。在此情况下的处理与第一实施例的步骤 S806 中所说明的处理相同。

[0125] 这样,可实现目标框架上的应用程序管理,就好像外观框架 402 上的应用程序被管理。

[0126] 第三实施例

[0127] 下面将参照附图详细说明根据本发明的第三实施例。注意,第三实施例中的装置和软件的结构与第一实施例中的相同,并将省略其说明。此外,可在不同的装置中准备外观框架和目标框架。

[0128] 图 12 示出第三实施例中使用的应用程序文件的格式。如图 12 所示,该应用程序文件将运行在包括外观框架 402 的多个目标框架上的应用程序合并为一个文件。该应用程序文件包括头区域 1201 和数据存储区域 1202。头区域 1201 包括表示应用程序文件的幻数 1211、表示头区域大小的头区域大小 1212、以及说明应用程序的各种属性的程序属性说明区域 1213。注意,程序属性说明区域 1213 包括图 13 所示的表示程序属性的 XML 数据。

[0129] 此外,数据存储区域 1202 被加密,使得只允许由外观框架 402 管理的密钥来解密。

当数据存储区域 1202 被解密时,其形成一系列程序块 1220,作为目标框架的应用程序。每个程序块 1220 包括该程序块 1220 的大小 1221、程序块标识符 1222、目标框架应用程序头区域 1223、以及程序数据存档区域 1224。目标框架应用程序头区域 1223 说明在感兴趣的目标框架上执行时所需的属性。程序数据存档区域 1224 存储将在目标框架上执行的程序数据。

[0130] 下面使用图 14 说明第三实施例中安装应用程序时执行的处理。假定外观框架 402 上的应用程序具有图 15 所示的 ApplicationAttribute 对象作为属性。

[0131] 图 14 是示出第三实施例中的安装处理的流程图。在步骤 S1401,与第一实施例中相同,执行下列处理。也就是说,当响应于操作者的指令接收到应用程序文件 520 时,外观框架 402 分析该应用程序文件 520 的头区域 1201 中的程序属性说明区域 1213 中所描述的 XML 文档。同时,外观框架 402 解密数据存储区域 1202。

[0132] 与图 13 所示的程序属性说明例子中的一样,使用“portion”标签来说明每个程序块 1220 的信息。要执行存储在程序块 1220 的程序数据存档区域 1224 中的应用程序的目标框架被说明为该 portion 标签中的目标属性。当没有设定目标属性时或者指定了外观框架 402 时,将外观框架 402 识别为目标框架。当程序块 1220 的标识符被指定为 portion 标签的 id 属性并且与程序块标识符 1222 进行比较时,对应于目标框架的程序块 1220 被指定。注意,所有的程序块信息被登记作为程序块列表。

[0133] 在步骤 S1402,使用步骤 S1401 中生成的程序块列表来处理每个程序块。从该列表中取出要被处理的程序块,并且如果还有待处理的程序块,则流程前进到步骤 S1403,以从程序块信息列表获得的程序块信息中获得目标框架信息。

[0134] 在步骤 S1404 中检查在步骤 S1403 中获得的目标框架信息是否被设定,即,外观框架 402 是否是目标。如果外观框架 402 是目标,则流程前进到步骤 S1405。在步骤 S1405,存储在感兴趣的程序块的程序数据存档数据区域 1224 中的程序被安装作为外观框架 402 的应用程序。此时,生成 ApplicationAttribute 对象来管理所安装的应用程序的应用程序属性。

[0135] 如果外观框架 402 不是目标,则流程前进到步骤 S1406。在步骤 S1406,执行与第一实施例的安装处理中的步骤 S804 中相同的处理,以获得感兴趣的目标框架的应用程序管理器。从该程序块信息中获得为了获得该应用程序管理器所需的信息。注意,可在不同的装置中准备外观框架 402 和目标框架。

[0136] 如果不能获得该应用程序管理器并且返回空,则流程前进到步骤 S1407。在步骤 S1407,表示没有与被指定为该应用程序的安装目的地的目标框架相对应的应用程序管理器被登记的警告页被显示在计算机 510 的 Web 浏览器上。该警告页通知操作者该安装已经失败。

[0137] 如果可获得该应用程序管理器,则流程前进到步骤 S1408,以在目标框架中安装。更具体来说,该程序块的程序数据存档区域 1224 中的数据作为输入流被打开,并且以该流作为变量调用该应用程序管理器的安装方法。当完成在该目标框架中的安装时,实现图 9 中所示的 Application 接口的对象被返回。该对象包括唯一识别安装在该目标框架中的应用程序的应用程序标识符。可通过调用该 Application 接口的 getApplicationID 方法来获得该应用程序标识符。

[0138] 如果应用程序列表没有被生成,则生成应用程序列表(作为 Java 的矢量对象)。由于在目标框架中的安装而返回的 Application 对象被登记在该应用程序列表中。

[0139] 另一方面,如果在步骤 S1402 中判断为没有剩余的程序块,则流程前进到步骤 S1409。如果在步骤 S1409 中判断为在步骤 S1405 中安装了外观框架 402 上的应用程序,则流程跳到步骤 S1411;否则,流程前进到步骤 S1410。

[0140] 在步骤 S1410,在外观框架 402 上生成虚拟应用程序,并生成应用程序信息对象。此时,生成 ApplicationAttribute 对象以管理应用程序属性。

[0141] 在步骤 S1411,将通过在步骤 S1408 中安装在目标框架中的应用程序的信息登记的应用程序列表设置在 ApplicationAttribute 对象的 frameworkList 字段中。当该应用程序在步骤 S1405 中被安装在外观框架 402 中或者在步骤 S1410 中与虚拟应用程序一起被安装时,生成该 ApplicationAttribute 对象。

[0142] 这样,当包括多个目标框架的应用程序的应用程序文件 520 被指定时,在各个目标框架中的安装被完成。此外,在该外观框架上生成该应用程序信息对象作为这些应用程序的代表。

[0143] 在第三实施例中,存储在一个应用程序文件 520 中的程序块作为相关的程序块来管理。然而,可根据相互无关的各个程序块生成虚拟应用程序,并且可将这些虚拟应用程序作为独立的虚拟应用程序来管理。

[0144] 下面说明安装以外的处理。安装以外的各个处理大致相同,作为代表性的例子,使用图 16 来说明停止处理。

[0145] 图 16 是示出第三实施例中应用程序的停止处理的流程图。与上述安装处理中相同,操作者从计算机 510 访问外观框架 402,以将外观框架 402 的应用程序组 403 的管理窗口显示在 Web 浏览器上。当操作者从所显示的管理窗口选择应用程序并指示进行停止处理时,流程前进到步骤 S1601。在步骤 S1601,从外观框架应用程序组 403 中的指定的应用程序中取出 ApplicationAttribute 对象,并获得该 ApplicationAttribute 对象的 frameworkList 字段的内容。此外,从 Vector 对象中取出 FrameworkProgram 对象。

[0146] 如果所有的 FrameworkProgram 对象都已经被取出,则流程前进到步骤 S1602,以在外观框架 402 的原始应用程序管理中执行应用程序停止处理。

[0147] 如果还有待取出的 FrameworkProgram 对象,则流程前进到步骤 S1603。在步骤 S1603,从在步骤 S1601 中取出的 FrameworkProgram 对象中获得作为目标框架标识符的 frameworkID 字段的值。此外,还获得与通过作为目标框架标识符的 frameworkID 字段的值选择的目标框架相对应的应用程序管理器。该获得方法与第一实施例中的相同,并将省略其说明。注意,可在不同的装置中准备该外观框架 402 和目标框架。

[0148] 然后在步骤 S1604 中检查是否可获得该应用程序管理器。如果不能获得该应用程序管理器,则生成异常,从而结束处理。如果可获得该应用程序管理器,则流程前进到步骤 S1605,以停止正在该目标框架上运行的所指定的应用程序。更具体来说,将存储在 FrameworkProgram 对象的 applicationID 字段中的该应用程序标识符指定为关于在步骤 S1603 中获得的应用程序管理器的变量,以调用停止方法。

[0149] 如上所述,可对存储在一个应用程序文件 520 中的不同框架上的应用程序进行生命周期管理。此外,可与外观框架 402 上的应用程序的生命周期管理同步地进行生命周期

管理。

[0150] 在上述实施例的每一个中,在一个应用程序文件 520 中存储应用程序是协作的条件。然而,分别安装的应用程序可以在安装之后被分组,以汇集为外观框架 402 上的应用程序,从而获得相同的效果。

[0151] 注意,本发明既可以应用于由多个装置(例如,主计算机、接口装置、读取器、打印机等)构成的系统,也可应用于由单个装置(例如,复印件、传真设备等)组成的设备。

[0152] 本发明的目的还可以通过向该系统或设备提供记录有能实现上述实施例的功能的软件程序的程序代码的存储介质,并且由该系统或装置中的计算机(或 CPU 或 MPU)读出并执行存储在该记录介质中的程序代码来实现。

[0153] 在这种情形下,从记录介质中读出的程序代码本身实现了上述实施例的功能,并且存储该程序代码的记录介质构成了本发明。

[0154] 作为用于提供该程序代码的记录介质,例如,可以使用软 (floppy®)盘、硬盘、光盘、磁光盘、CD-ROM、CD-R、磁带、非易失性存储卡、ROM 等。

[0155] 不仅可以通过计算机执行所读出的程序代码来实现上述实施例的功能,还可以基于该程序代码的指令通过在计算机上运行的 OS(操作系统)执行部分或全部的实际处理操作来实现上述实施例的功能。

[0156] 此外,还可以通过将从该记录介质中读出的程序代码写入插入或连接到计算机的功能扩展板或功能扩展单元的存储器中之后,由设置在该功能扩展板或功能扩展单元中的 CPU 等执行部分或全部实际处理操作来实现上述实施例的功能。

[0157] 由于在不脱离本发明的精神和范围的情况下,可做出许多明显不同的实施例,因此应当理解,除了由所附权利要求限定外,本发明不局限于具体实施例。

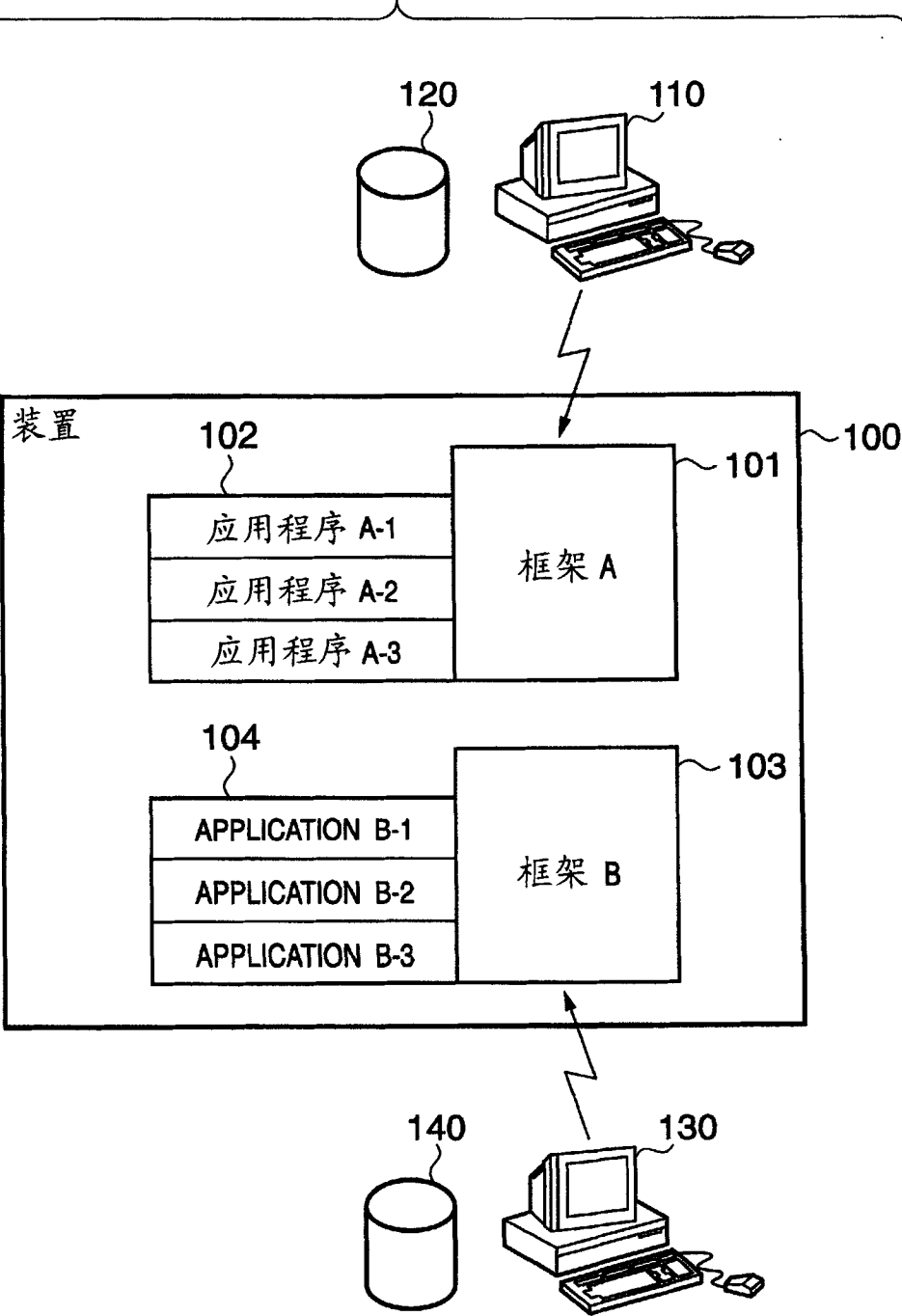


图 1

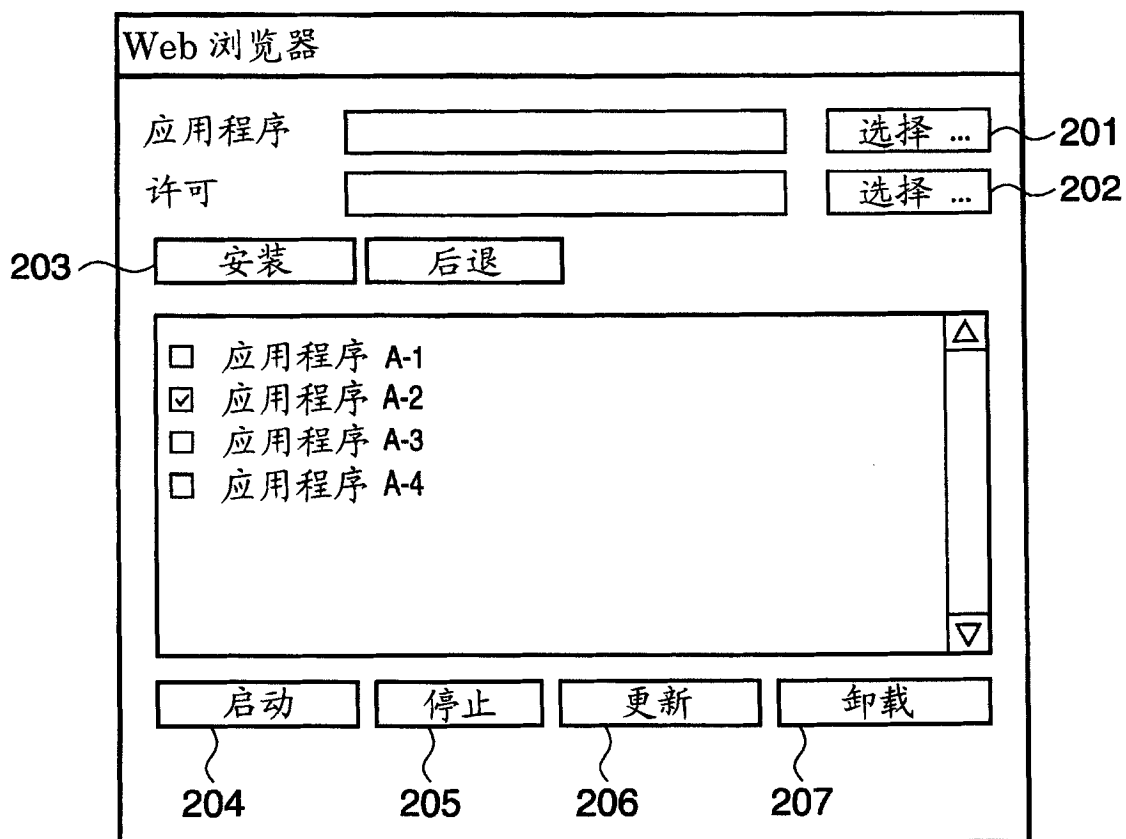


图 2

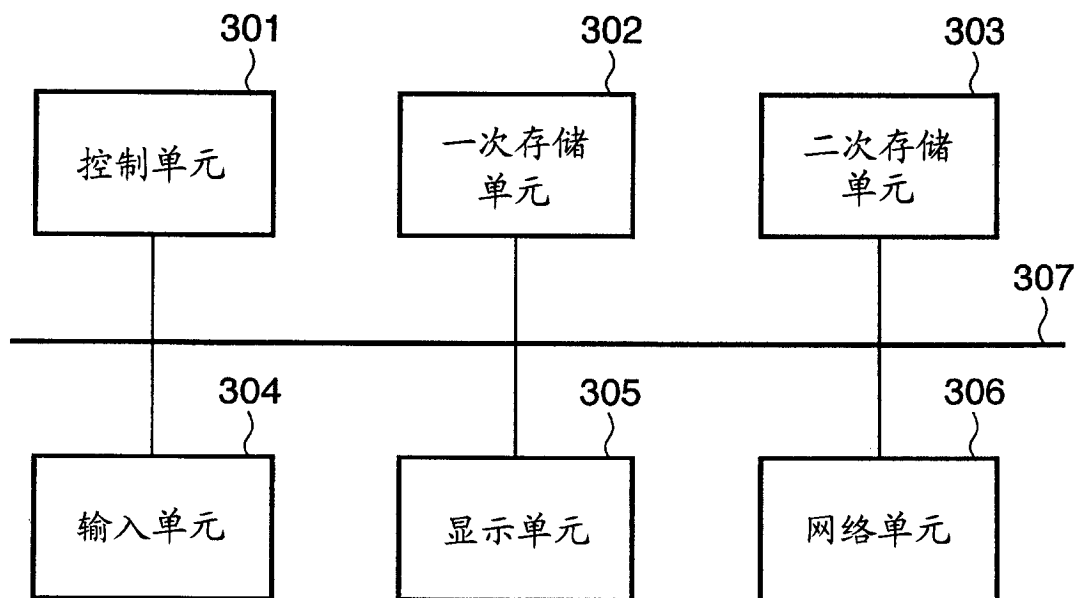


图 3

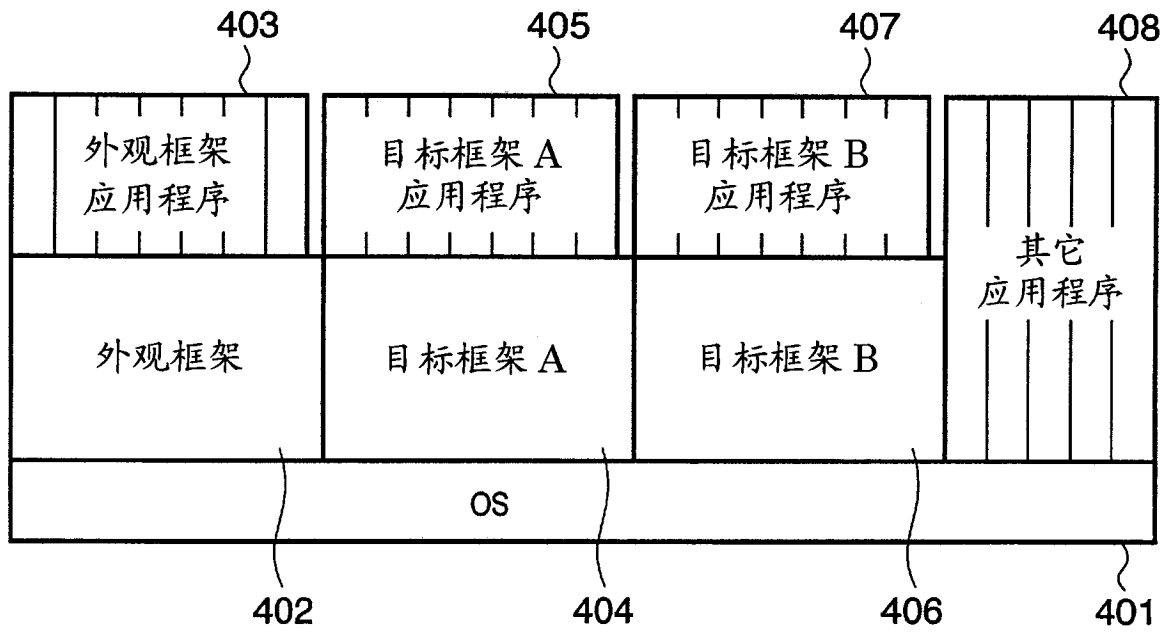


图 4

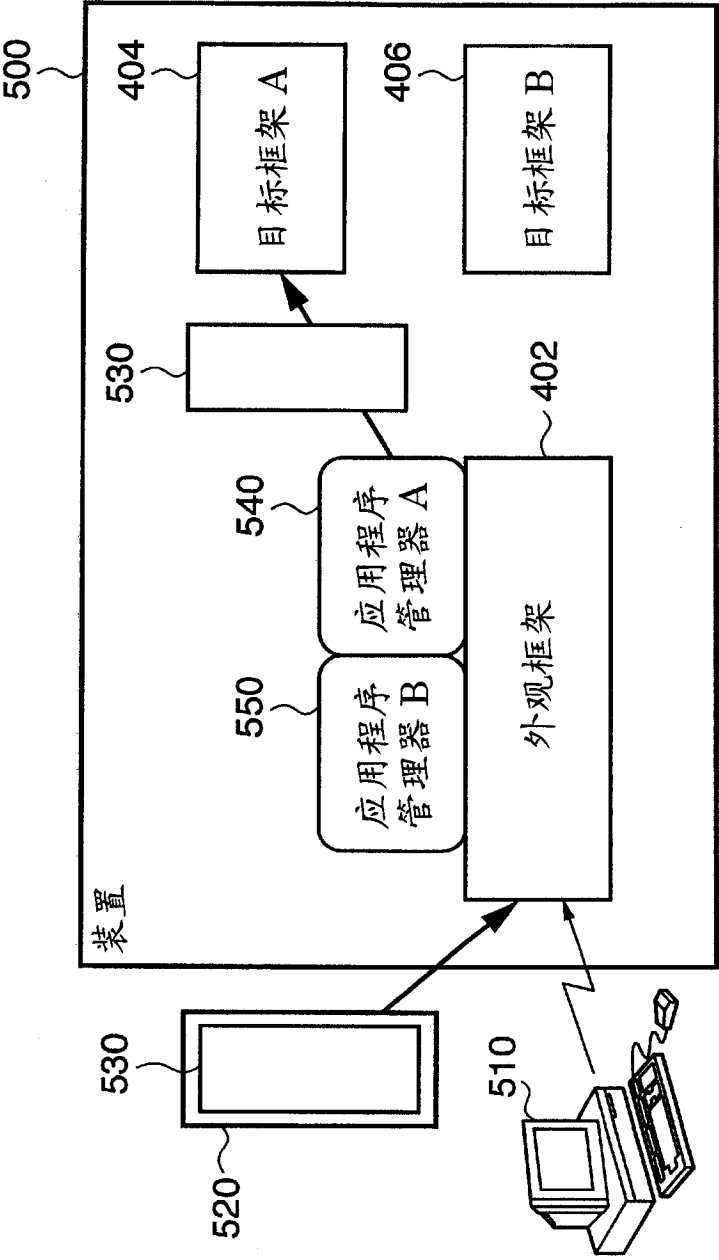


图 5

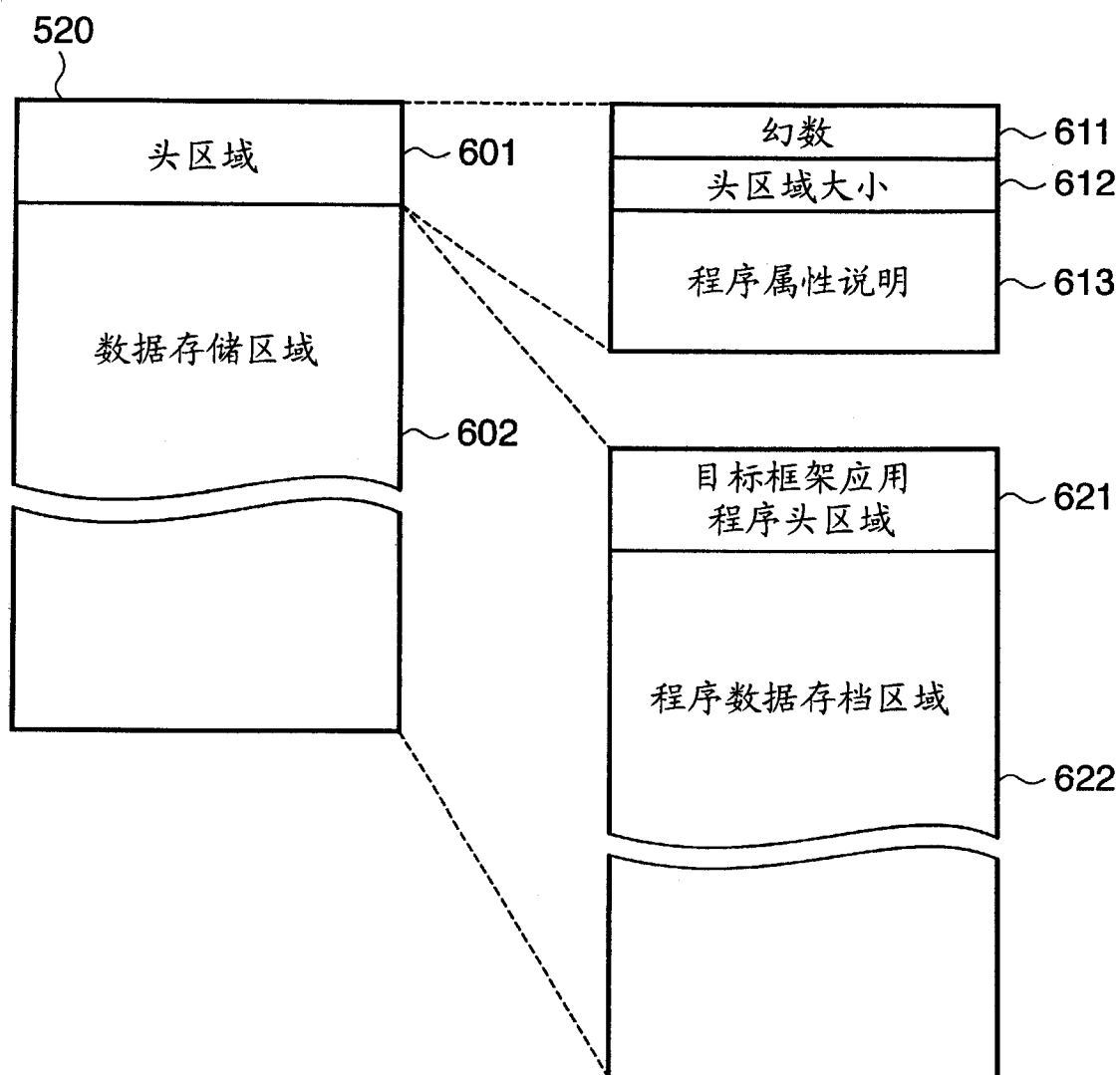


图 6

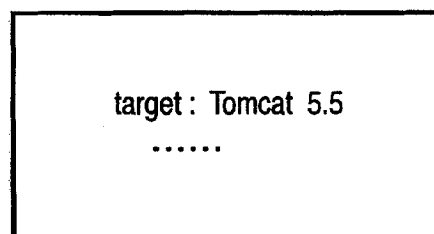


图 7

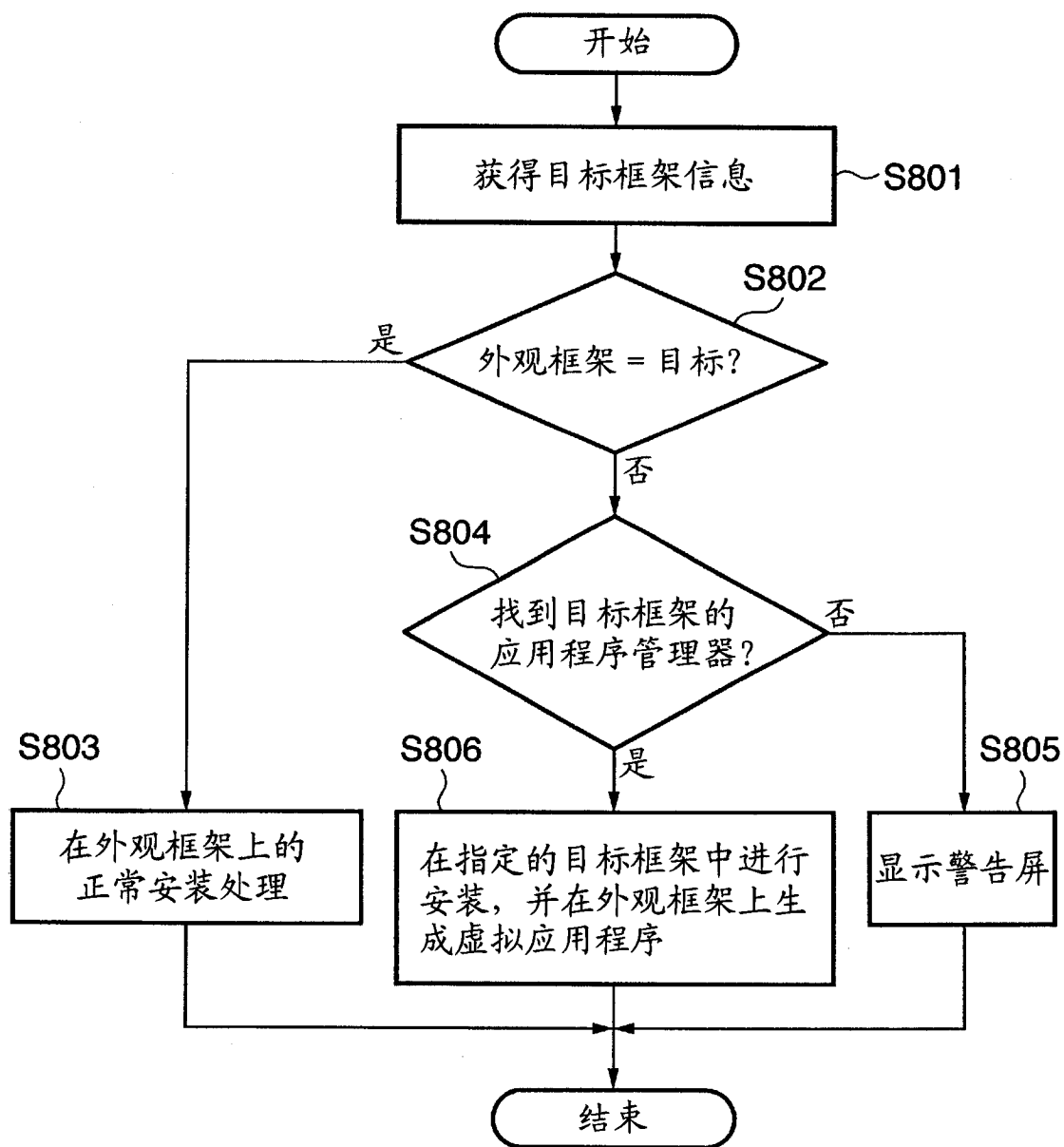


图 8

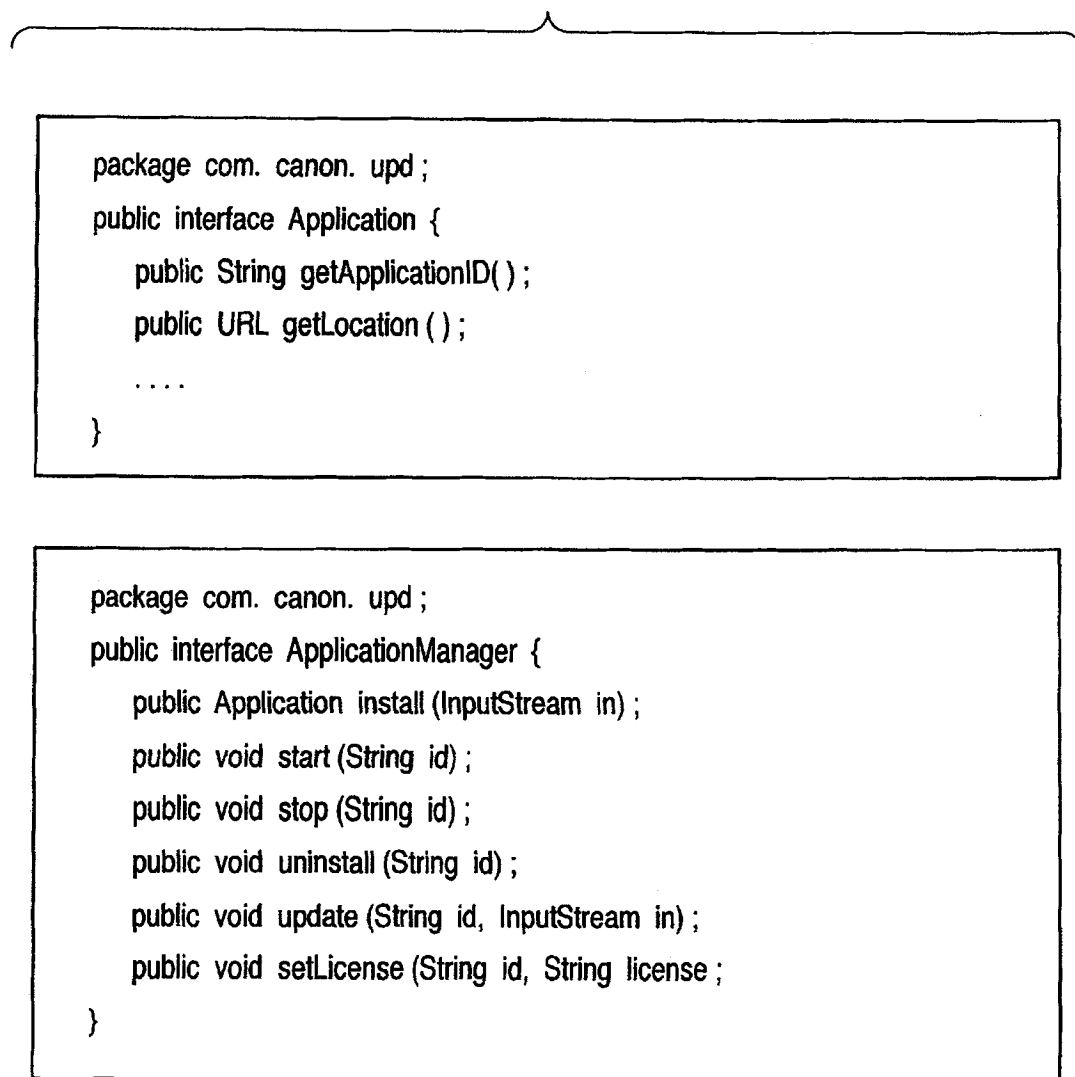


图 9

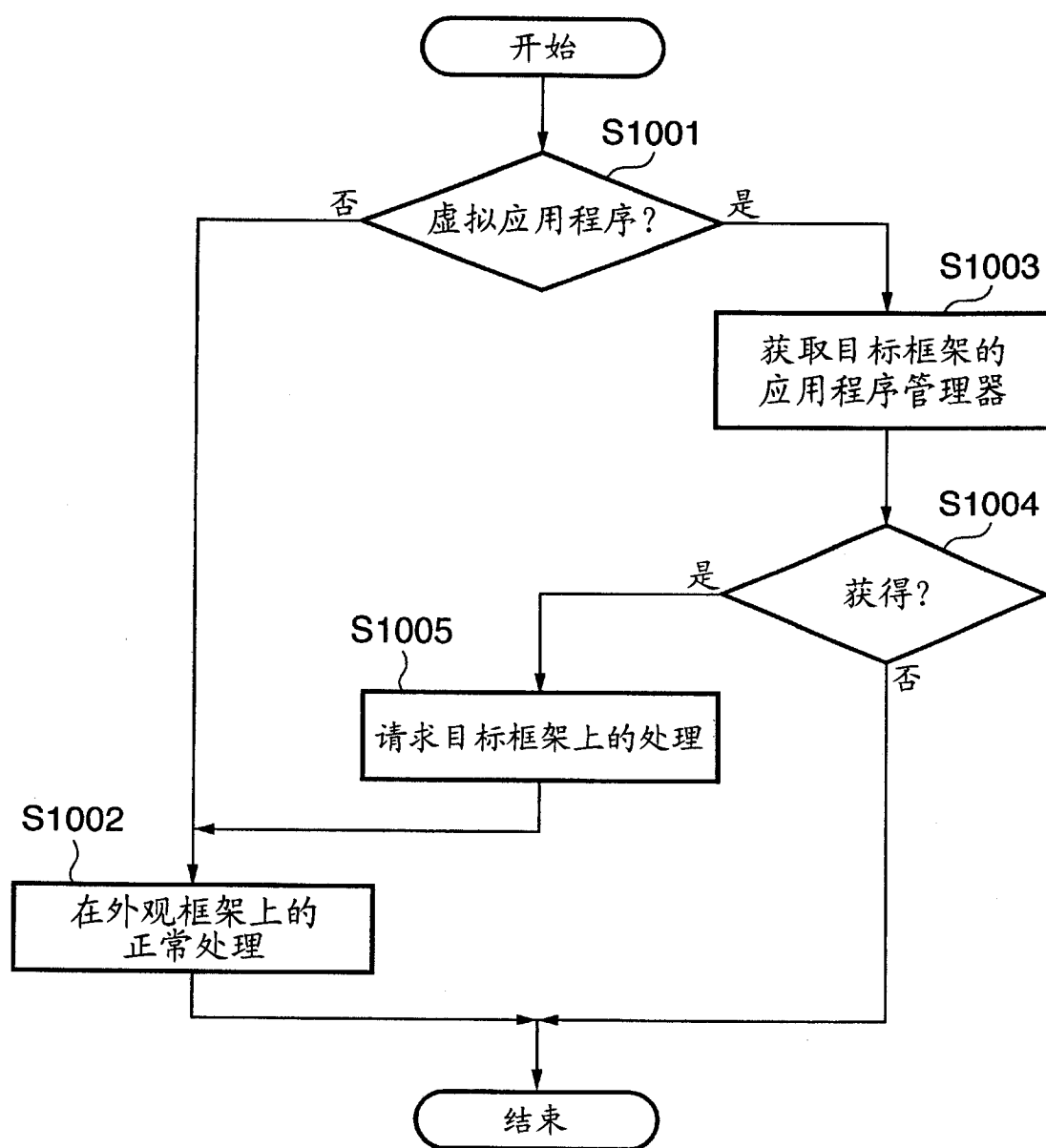


图 10

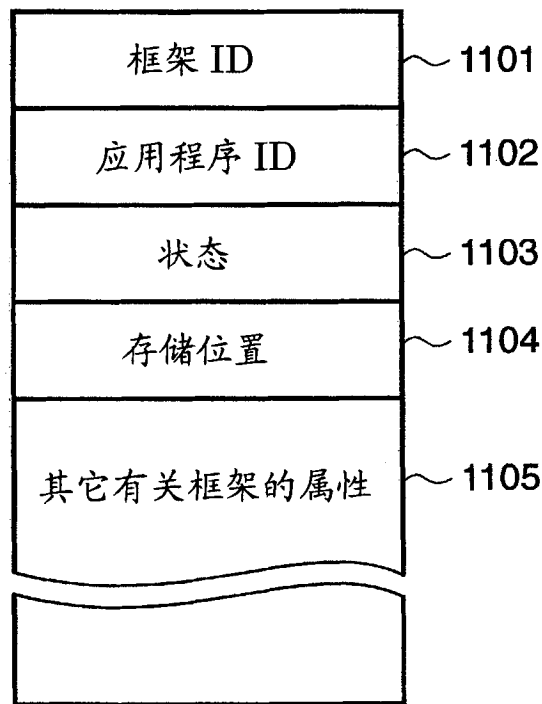


图 11

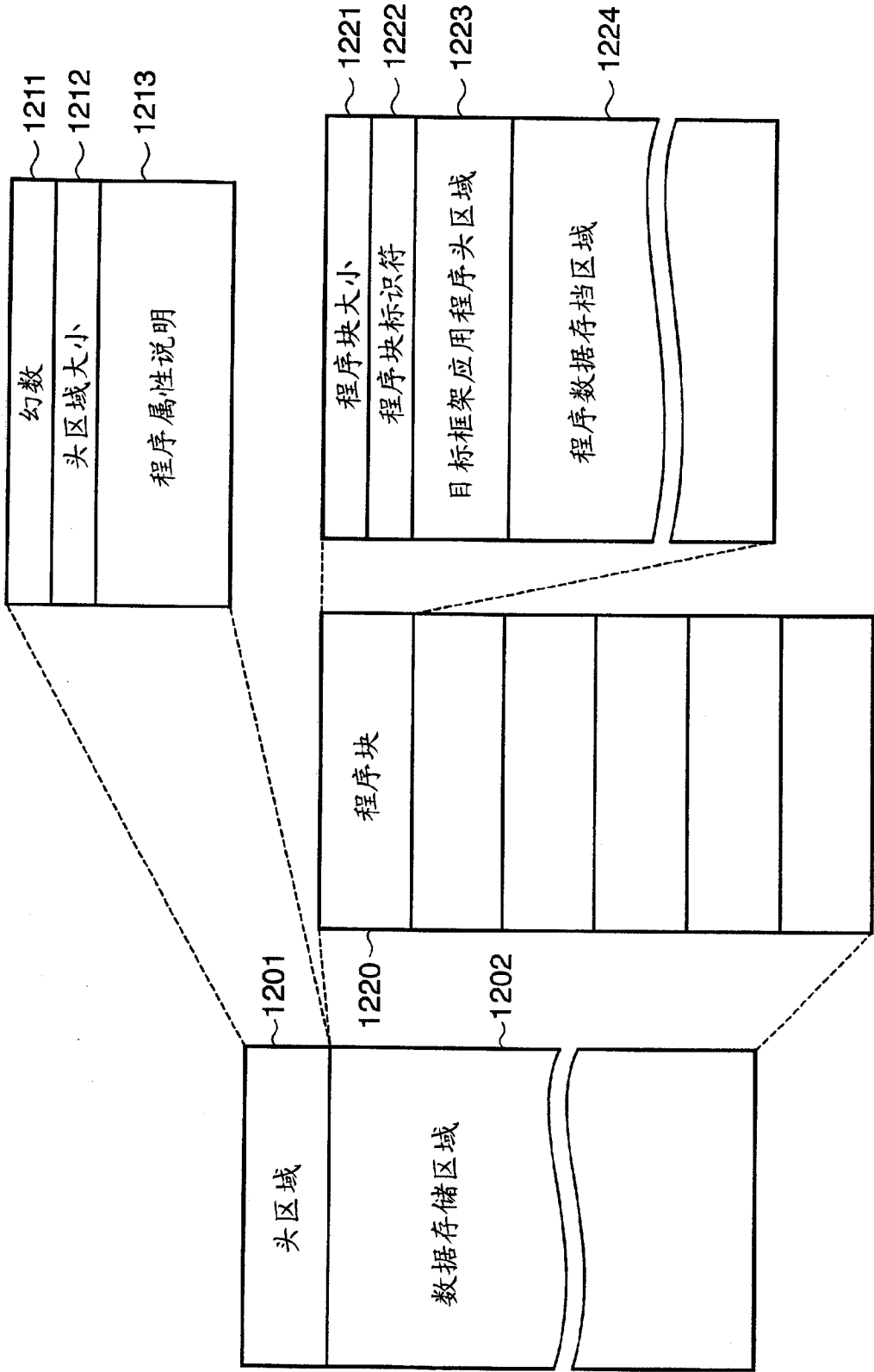


图 12


```
<?xml version="1.0" encoding="utf-8" ?>  
<application>  
  <portion target="Tomcat 5.5" id="portion1">  
  </portion>  
  .....  
</application>
```

图 13

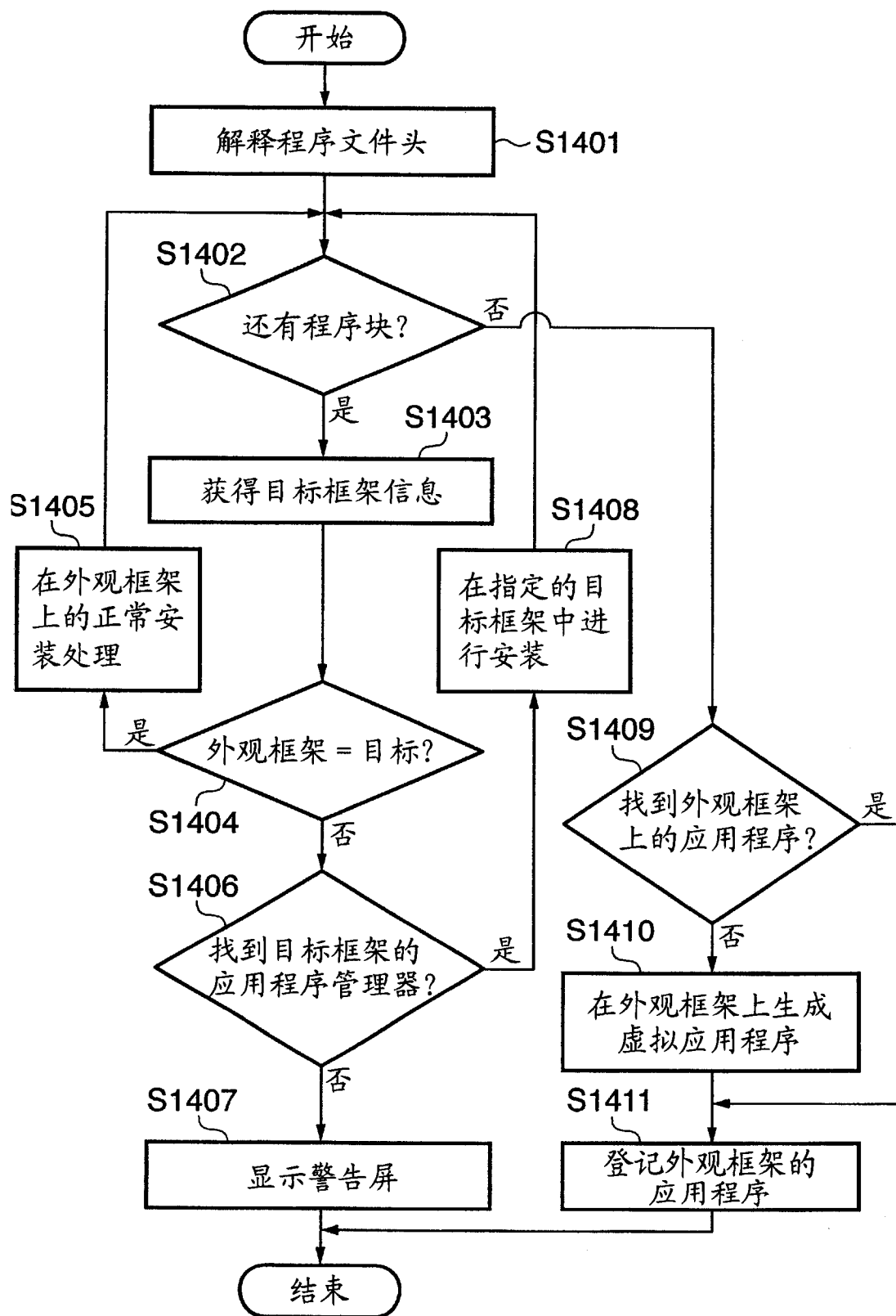


图 14

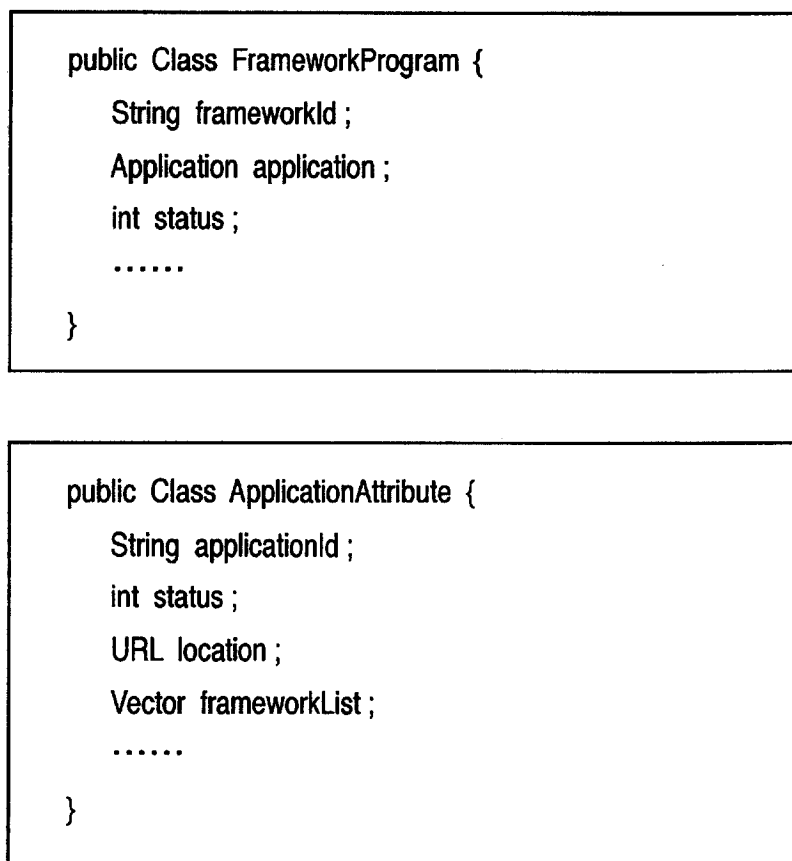


图 15

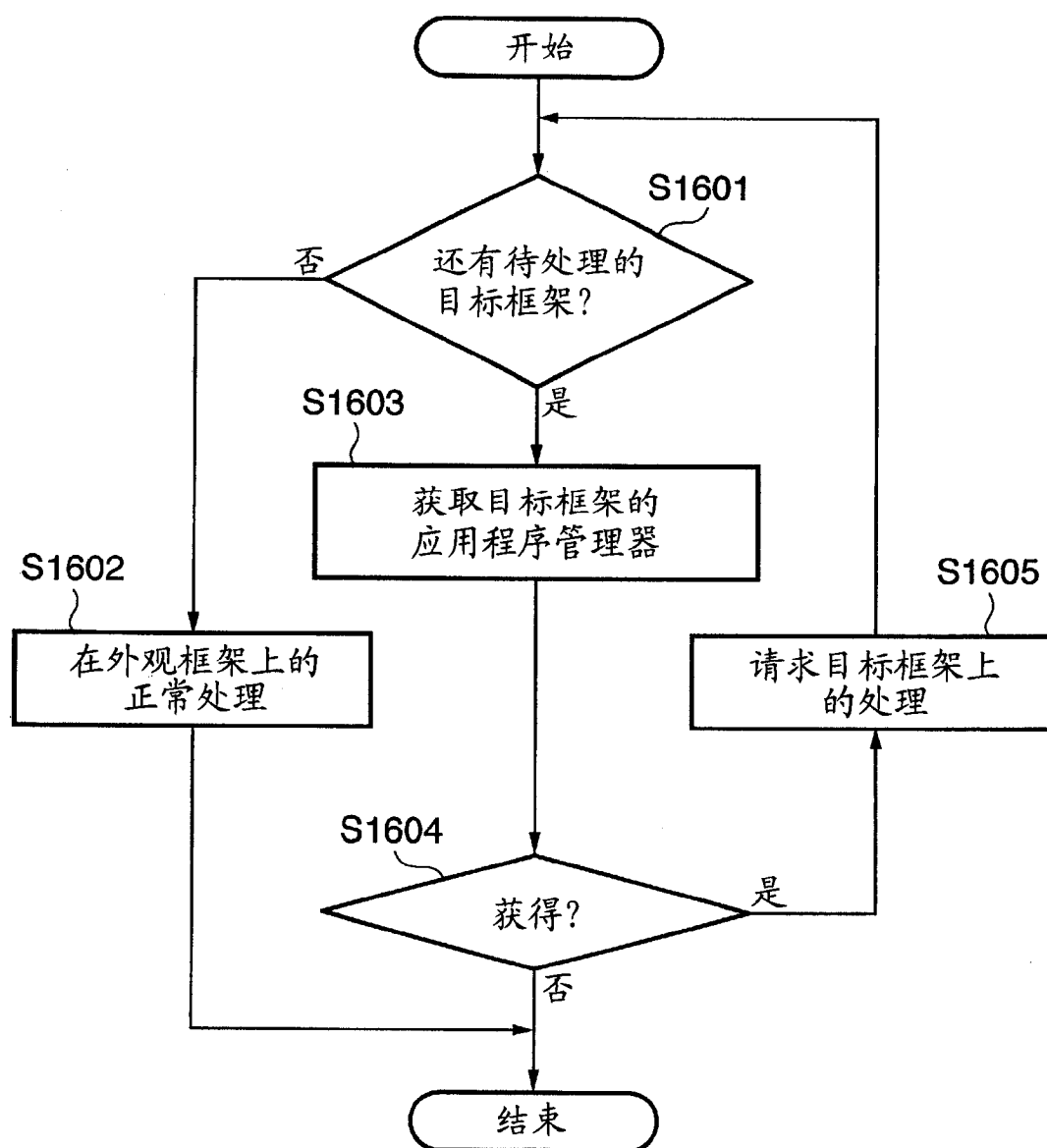


图 16