



(12) 发明专利申请

(10) 申请公布号 CN 101978366 A

(43) 申请公布日 2011. 02. 16

(21) 申请号 200980110662. 1

代理人 胡利鸣 钱静芳

(22) 申请日 2009. 02. 20

(51) Int. Cl.

(30) 优先权数据

G06F 15/16 (2006. 01)

12/052, 550 2008. 03. 20 US

G06F 9/44 (2006. 01)

G06F 3/00 (2006. 01)

(85) PCT申请进入国家阶段日

2010. 09. 19

(86) PCT申请的申请数据

PCT/US2009/034601 2009. 02. 20

(87) PCT申请的公布数据

W02009/117207 EN 2009. 09. 24

(71) 申请人 微软公司

地址 美国华盛顿州

(72) 发明人 O·阿米加 J·麦金太尔

D·舒克拉 J·张

(74) 专利代理机构 上海专利商标事务所有限公

司 31100

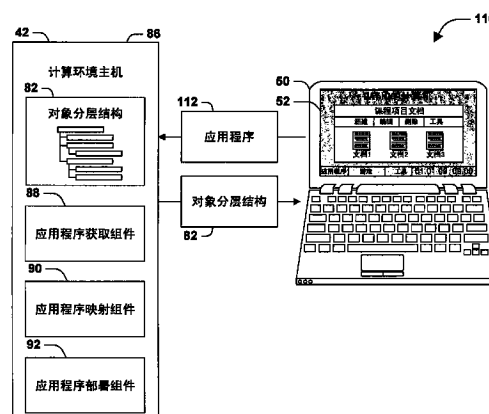
权利要求书 2 页 说明书 17 页 附图 10 页

(54) 发明名称

可部署对象分层结构内的应用程序管理

(57) 摘要

对计算环境中的应用程序的管理通常涉及大量计算操作,如获取应用程序、在各个位置部署数据对象、以及相对于所部署的设备来配置应用程序。常规计算环境可能在整个典型的应用程序生存周期中提供不足够的支持,尤其是对跨许多设备分布的计算环境而言。相反,计算环境可以被表示在可被分发给各设备的可部署对象分层结构中。因此可以提供应用程序管理服务以将应用程序的表示安装在对象分层结构中,以及支持应用程序生存周期的许多其他方面,如应用程序发现、共享、更新、以及移除。这些设备因此可以用一致的方式向用户提供计算环境的应用程序,同时减少了构成计算网络的设备之间的手动管理。



1. 一种用于对将由应用程序存储 (84) 管理的至少一个应用程序 (112) 部署到表示至少一个设备 (50) 上的计算环境的对象分层结构 (44) 进行管理的系统 (86), 所述系统 (86) 包括:

应用程序获取组件 (88), 所述组件被配置成:

接收安装应用程序 (112) 的请求, 以及

从所述应用程序存储 (84) 获取所述应用程序 (112);

应用程序映射组件 (90), 所述组件被配置成将所述对象分层结构 (44) 中表示的至少一个设备 (50) 映射到至少一个已安装应用程序 (112); 以及

应用程序部署组件 (92), 所述组件被配置成将所述至少一个已安装应用程序 (112) 部署在所述对象分层结构 (44) 中。

2. 如权利要求 1 所述的系统, 其特征在于, 所述系统包括表示至少一个应用程序的应用程序存储, 各个应用程序包括至少一个应用程序资源。

3. 如权利要求 1 所述的系统, 其特征在于, 包括:

被配置成存储所述对象分层结构的对象分层结构存储, 以及

被配置成将所述对象分层结构的至少一部分发送到所述至少一个设备的对象分层结构传输组件。

4. 如权利要求 1 所述的系统, 其特征在于, 包括:

应用程序发现组件, 所述组件被配置成:

从用户接收应用程序查询,

选择满足所述应用程序查询的应用程序, 以及

将所选应用程序呈现给所述用户。

5. 如权利要求 1 所述的系统, 其特征在于, 所述应用程序获取组件被配置成:

通过所述设备将应用程序许可证呈现给用户, 以及

在从所述用户接收到对所述应用程序许可证的接受后, 记录对所述应用程序许可证的接受。

6. 如权利要求 1 所述的系统, 其特征在于, 所述应用程序获取组件被配置成对与所述至少一个设备的用户的获取事务进行中介。

7. 如权利要求 1 所述的系统, 其特征在于, 所述应用程序获取组件被配置成生成表示所述应用程序的用户的用户帐户。

8. 如权利要求 1 所述的系统, 其特征在于, 所述部署包括:

相对于所述设备和所述设备的用户中的至少一个来配置所述应用程序。

9. 如权利要求 1 所述的系统, 其特征在于, 包括:

被配置成在所述至少一个设备上启动所述应用程序的应用程序启动组件。

10. 如权利要求 1 所述的系统, 其特征在于, 包括:

应用程序共享组件, 所述组件被配置成:

接收与一用户共享所述应用程序的请求, 以及

在接收到共享所述应用程序的请求后, 与所述用户共享所述应用程序。

11. 如权利要求 10 所述的系统, 其特征在于, 所述应用程序共享组件被配置成与所述用户共享所述对象分层结构中包括以下至少一个的一部分: 所述应用程序、应用程序会话、

应用程序配置、以及所述应用程序生成的至少一个数据对象。

12. 如权利要求 1 所述的系统,其特征在于,包括:

被配置成更新所述应用程序的应用程序更新组件。

13. 如权利要求 12 所述的系统,其特征在于,所述应用程序更新组件被配置成更新所述对象分层结构中的至少一个应用程序资源,以及

被配置成撤消所述应用程序的应用程序撤消组件。

14. 如权利要求 1 所述的系统,其特征在于,包括:

被配置成请求撤消所述应用程序的接收组件。

15. 如权利要求 14 所述的系统,其特征在于,所述接收组件被配置成在所述对象分层结构中撤消所述应用程序。

可部署对象分层结构内的应用程序管理

[0001] 背景

[0002] 许多计算环境都包括由一组对象系统管理的大型且不同的对象集合。例如，计算环境可包括由文件系统管理的文件集合、由数据库系统管理的一个或多个数据库、表示应用程序的并由汇编高速缓存管理的可执行二进制代码集合、由操作系统的用户简档组件管理的用户简档集合、以及由数据高速缓存组件管理的各种数据高速缓存，如由操作系统的复制缓冲区管理的复制项的集合、由操作系统的撤消组件管理的撤消动作的集合、以及由操作系统的项使用跟踪组件管理的“最近使用”项的集合。此外，可以在由一个或多个用户操作的多个设备之间并且根据一个或多个操作（例如，合并两个对象集合的对象同步操作和调整目标对象集合以与源对象集合进行匹配的对象镜像操作）来交换这些对象。以此方式，这些对象通过一组对象系统松散地进行组织并聚集以表示计算环境。

[0003] 常规计算环境内的应用程序管理通常涉及相对大的用户参与。例如，用户可独立地发现所需应用程序，并可以取得安装程序包，如安装可执行码或包含安装文件的 CD-ROM。用户随后可以利用便于将应用程序部署在计算环境中（如通过将文件复制到特定位置并创建注册信息）的应用程序安装程序。然而，在应用程序部署之后，计算环境通常向该应用程序提供很少的进行中支持，有时只包括卸载程序。诸如执行、共享、同步、以及更新等常见维护任务通常移交给应用程序。此外，应用程序在一个计算机系统或设备上的部署很少允许或便于该应用程序在其他计算机系统或设备上的部署，即使由同一用户操作并且在两个或多个这样的平台上用于共同目的。

[0004] 概述

[0005] 提供本概述是为了以简化的形式介绍将在以下详细描述中进一步描述的一些概念。本概述并不旨在标识出所要求保护的主题的关键因素或必要特征，也不旨在用于限定所要求保护的主题的范围。

[0006] 将对象在计算系统内组织成不相交的松散聚集的对象系统集合可以造成若干问题。例如，可能难以通过各种设备向用户呈现一致的计算环境，尤其是在这些设备能力不同的情况下（例如，高性能个人工作站、基于浏览器的公共终端、以及低性能蜂窝电话设备）。作为另一示例，将诸如同步和备份等服务应用于对象可涉及与每一对象系统进行接口以影响由此管理的对象，并且这样的接口可以随对象系统而变化。作为第三示例，使不同的一组对象相关（如，构成一应用程序的所有对象）可能因管理对象系统的不完全协作而是困难的。

[0007] 用于表示构成计算环境的对象的一种替换技术涉及将这些对象组织成可由计算环境主机来主存的对象分层结构。如果对象是由对象系统用统一方式来表示的并以一致的方式来管理的，则可以设计一组服务来应用于计算环境的所有对象。此外，对象分层结构可被传递到各个设备以表示同一计算环境（包括相同的用户简档、应用程序、数据文件等），并且每一设备可以用一致的方式来呈现该计算环境，但基于该设备的能力（例如，用于从附连到工作站的键盘设备接收数据输入的硬键盘接口，以及用于从蜂窝电话设备接收数据输入的触摸屏软键盘接口）进行了定制。

[0008] 在这一性质的对象分层结构内,可以执行应用程序管理任务。计算环境主机可以便于获取应用程序、将该应用程序与一设备相关联、以及将该应用程序部署在该设备上。将应用程序表示成可部署对象分层结构中的一组对象可便于常见的应用程序管理任务。作为第一示例,并且在许多自动部署的场景中,应用程序可以容易地部署在该对象分层结构中表示的多个计算机系统和设备上。例如,该应用程序的第一版本可以部署到工作站;该应用程序的第二版本可以部署成用于公共终端的 web 应用程序或启用 web 的应用程序;并且该应用程序的第三版本可以部署成供在低功率蜂窝电话设备上使用的减少功能的应用程序。作为第二示例,计算环境主机可以便于通过相应地调整对象分层结构并通过将所主存和部署的该对象分层结构的各版本进行同步来将这些调整部署到各设备以更新应用程序(例如,使用更新版本对应用程序资源进行修补或部署供与该应用程序一起使用的新内容)。

[0009] 此外,将应用程序表示成可部署对象分层结构中的一组对象还可启用新特征。作为第一示例,通过将对象分层结构的特定部分部署到第二用户的设备、计算机系统、或计算环境主机,计算环境可使用户能够与第二用户共享应用程序或应用程序会话的一方面。例如,第一用户可以选择一组歌曲对象来在第一计算机上通过媒体播放器应用程序进行播放,并可选择与第二用户共享该媒体播放器应用程序会话(例如,通过将媒体播放器应用程序的音频流传输到第二用户)。作为第二示例,应用程序的配置可以跨各设备和计算机系统来作为对象分层结构的一部分进行同步。设备无关应用程序属性(例如,添加到拼写检查字典的自定义单词)可以自动地传播到所有这样的设备和计算机系统,而计算环境主机可为相应设备指定设备专用应用程序属性。以此方式,计算环境主机的应用程序管理服务可以支持应用程序在由一个或多个用户所操作的各计算机系统和设备上的发现、部署、使用 and 移除。

[0010] 为实现上述和相关目的,以下描述和附图阐述了各个说明性方面和实现。这些方面和实现仅指示可使用一个或多个方面的各种方式中的一些。结合附图阅读以下详细描述,则本发明的其他方面、优点、以及新颖特征将变得显而易见。

[0011] 附图描述

[0012] 图 1 是计算环境内的典型应用程序生存周期的图示。

[0013] 图 2 是计算环境主机与各计算环境呈现设备的示例性交互的图示。

[0014] 图 3 是示出对将应用程序部署在表示设备上的计算环境的对象分层结构中进行管理的示例性方法的流程图。

[0015] 图 4 是示出用于对将由应用程序存储管理的至少一个应用程序部署到表示至少一个设备上的计算环境的对象分层结构进行管理的示例性系统的组件框图。

[0016] 图 5 是示出用于对将应用程序部署到表示至少一个设备上的计算环境的对象分层结构进行管理的另一示例性系统的组件框图。

[0017] 图 6 是示出用于对将应用程序部署到表示至少一个设备上的计算环境的对象分层结构进行管理的又一示例性系统的组件框图。

[0018] 图 7 是示出用于对将应用程序部署到表示至少一个设备上的计算环境的对象分层结构进行管理的示例性系统的一部分的组件框图。

[0019] 图 8 是示出用于对将应用程序部署到表示至少一个设备上的计算环境的对象分层结构进行管理的示例性系统的另一部分的组件框图。

[0020] 图 9 是示出用于对将应用程序部署到表示至少一个设备上的计算环境的对象分层结构进行管理的示例性系统的又一部分的组件框图。

[0021] 图 10 是示出用于对将应用程序部署到表示至少一个设备上的计算环境的对象分层结构进行管理的又一示例性系统的组件框图。

[0022] 图 11 示出了其中可实现此处所阐述的原理中的一个或多个的示例性计算环境。

[0023] 详细描述

[0024] 现在参考附图来描述所要求保护的主体,所有附图中使用相同的附图标记来指代相同的元素。在以下描述中,为解释起见,阐明了众多具体细节以提供对所要求保护的主体全面理解。然而,很明显,所要求保护的主体可以在没有这些具体细节的情况下实施。在其他情况下,以框图形式示出了公知的结构和设备以便于描述所要求保护的主体。

[0025] 常规计算环境通常包括大量各种各样的一批应用程序,范围从具有简单目的的小工具(如时钟和计算器)到复杂且相互关联的大型应用程序套件(如软件集成开发环境和生产力套件)。应用程序还可包括媒体,如多媒体播放器和由此呈现的媒体对象(例如,要由阅读器应用程序呈现的一本或多本电子书)、游戏、屏幕保护程序、用于管理各种类型的硬件的设备驱动程序,等等。应用程序可以部署在本地(例如,通过部署到用户的设备的可执行二进制码和资源)、远程(例如,在远程服务器上运行并通过 web 浏览器与用户进行通信的 web 应用程序)、或其组合(例如,利用远程资源的、安装在本地的应用程序,如被配置成对网络可访问电子邮件档案进行访问的电子邮件客户端)。通常提供操作系统以主存和维护应用程序并通过应用程序启动程序和与同它相关的数据对象的关联来向用户提供对该应用程序的访问。

[0026] 图 1 示出计算环境中的应用程序所展示的典型应用程序生存周期 10。在典型计算环境中表现的应用程序生存周期 10 可涉及大量操作,如复制数据对象、写入和读出配置信息、以及通过计算机网络向服务器请求并从中接收对象。应用程序生存周期 10 通常以取得 12 开始,其中用户选择要使其可在计算环境中访问的应用程序。取得 12 通常包括发现 14,其中用户变得知道该应用程序,如通过用户驱动的对应用程序库的浏览或广告建议。取得 12 通常还包括获取 16,其中用户获得该应用程序的资源并安排用于使用该应用程序的许可。对于安装在本地的应用程序,获取 16 可以涉及从应用程序服务器下载安装程序应用程序;而 web 可访问应用程序可涉及与该 web 应用程序的提供者签约,如通过创建用于 web 邮件服务的用户帐户。获取 16 还可涉及交易(如支付金钱)和许可(如,向用户呈现最终用户许可协议(EULA)并请求接受其条款和条件)。取得 12 的部分或全部有时可以省略,如通过使预先安装的应用程序可作为新计算环境的默认组件对用户可用。

[0027] 在取得 12 之后或与其同时,应用程序生存周期 10 通常涉及安装 18,其中本地计算环境被配置成提供对该程序的访问。安装 18 有时涉及映射 20,其中特定设备被配置成使用该应用程序。映射 20 可涉及用户指定用于访问该软件的特定设备(例如,选择要在其上部署应用程序的蜂窝电话设备)和/或向设备授权使用该软件(例如,通过产品激活)。安装 18 还涉及部署 22,其中将构成该应用程序的资源部署在一个或多个映射到的设备上(例如,创建应用程序文件夹、将应用程序二进制码置于这些文件夹中、注册代码库、以及将配置信息写入系统注册表和配置文件)。部署 22 还可涉及配置计算环境以提供对该应用程序的访问,如通过将该应用程序向应用程序启动程序注册、将该应用程序与同其相关的数

据对象进行关联（例如，将一媒体应用程序标识为特定类型的媒体对象的媒体播放器）、和/或配置计算环境以在系统初始化时或响应于特定事件来自发地启动该应用程序。部署 22 还可涉及显示文档编制以帮助用户使用该应用程序。在一些场景中，安装 18 可涉及配置远程资源，如将设备向网络服务器注册或配置网络服务器来为该设备主存该应用程序。在其他场景中，安装 18 可被减少或省略；例如，部署在本地的应用程序可包括可在不配置或安装的情况下运行的可执行二进制码，并且 web 应用程序可能不涉及在本地设备上的任何类型的安装。

[0028] 在取得 12 和安装 18 之后，应用程序可做好准备来由一个或多个用户进行使用 24。使用 24 通常涉及执行 26，它例如可通过启动该应用程序或激活与该应用程序相关联的数据对象来开始。执行 26 还可包括获取附加内容，如可由媒体播放器呈现的新媒体对象。使用 24 还涉及更新 28，它包括对该应用程序的进行中改变，如修补以提供该应用程序的新版本或对由此利用的资源进行维护（例如，接收并合并用于检测并解决新形式的恶意软件的信息的抗病毒包）。这样的更新 28 有时可以由应用程序服务器对远程资源执行；例如，web 邮件提供者可以在不涉及用户或用户设备并且甚至在不知道用户的情况下频繁更新其后端软件。

[0029] 应用程序生存周期 10 还可涉及卸载 30，其中从一个或多个设备移除该应用程序。卸载 30 可包括解除映射 32，其中解除一个或多个设备使用该应用程序的授权。例如，web 应用程序只可在某些设备上访问，并且可以通过将设备从存储在应用程序服务器上的授权设备列表中移除来解除映射。卸载 30 还可包括退役 34，其中从一个或多个设备移除应用程序的资源。退役 34 可由用户来请求，或可以自发地调用（例如，可以只在有限时间段内将应用程序许可给用户，并且计算环境可被配置成在该许可证期限期满之后自动地使该应用程序退役）。

[0030] 虽然对各种各样的应用程序广泛地使用了应用程序生存周期，但许多常规计算环境被配置使得应用程序生存周期展示出若干缺点。作为一个示例，常规计算环境可能需要用户大量参与进行该应用程序生存周期。在取得 12 期间，计算环境可能对应用程序的发现或获取提供很少支持或不支持，并且用户可能承担了通过研究（例如，web 搜索）寻找有用的应用程序并下载应用程序安装包的任务。取得 12 还可涉及用户接受许可证，但这有时可能被废除（例如，用户可以预先授权在特定样板最终用户许可协议下所提供的任何应用程序）。现代计算环境通常对映射 20 提供很少支持或不支持，并且用户可能必须安排应用程序在一批设备上的单独部署。许多常规计算环境提供对部署 22 的一些支持，如自动化安装程序工具。然而，这样的安装程序工具仍然可能向用户委托可以其他方式避免的相当大量的交互，如在 web 应用程序中创建用户帐户时输入用户信息（名字、电子邮件地址等）、提供信用卡信息以购买许可证、以及输入诸如安装 CD 等物理介质包括的序列密钥。常规计算环境也对更新 28 提供不充分的支持，它通常被委托给应用程序或甚至用户启动的发现、下载、以及应用软件修补。最后，解除映射 34 和退役 34 可由安装程序应用程序来促进，但可被限于按设备管理；例如，用户可能必须在每一设备上启动卸载并且所有这些设备之间的集中式管理的部署可能是困难的。

[0031] 在现代计算环境使用（其通常扩展至一批设备）的上下文中，这些缺点可能恶化并且可能产生其他缺点。计算环境的用户可协作地管理并利用工作站、便携式计算机系统，

如笔记本或平板计算机、游戏控制台、蜂窝电话、以及媒体播放器。此外,用户可能希望访问公共工作站上的计算环境的全部或部分,或向其他用户授予对计算环境的部分或全部的访问权(例如,以允许顾客利用计算环境或访问其中管理的数据对象)。尽管用户可在计算环境的上下文中操作的设备持续增长,但常规操作系统可能对计算环境在这些设备之间的一致呈现提供很少支持,并且由此对促进数据对象的分发、访问、以及同步提供很少支持。这一有限支持对应用程序尤其明显。一般而言,用户通常在每一设备上独立于任何其他设备的配置来手动地执行应用程序生存周期 10 的许多方面。例如,用户可以根据用户的偏好来配置特定应用程序,但在每一设备上以类似方式配置应用程序通常涉及每一应用程序部署的手动配置。另外,以协作方式使用多个设备可以造成新问题;例如,用户可能在第一设备上具有关于一个人的一组联系人信息,该联系人信息与同一个人第二设备上的一组联系人信息不一致,并且可能不能够查明哪一组联系人信息是更新的。

[0032] 最近,作出了尝试来开发用于以一致的、可部署的且可扩展的方式来提供对一批设备之中的计算环境的访问的技术。这些技术还设法在进行协作的设备之间提供共同的应用程序集合,以及用于管理应用程序在这些设备之间的取得、安装、使用、和卸载的集中式服务。该应用程序集合在各设备之间不一定相同;例如,工作站可包含可能不能在蜂窝电话设备上良好地运行的高性能应用程序(例如,照片编辑软件和图形密集游戏),而蜂窝电话设备可包括与非便携工作站不相关的便携性应用程序(例如,基于GPS的地图软件)。然而,许多应用程序和与其相关的数据对象可以在这些设备之间共享(例如,被配置成管理用户日历对象的日历应用程序),并且该计算环境可以适应于能够在这些设备之间分发和同步该应用程序和数据对象。该应用程序还可以针对每一这样的设备来进行适应;例如,该应用程序的第一版本或配置可以面向诸如工作站等高性能设备,而该应用程序的第二版本或配置可以面向具有有限屏幕大小和输入能力的低性能设备。然而,同一应用程序的应用程序配置的其他部分(例如,用户偏好,如视觉样式)可以在设备之间共享,并可以促进用户对该计算环境在各设备之间的体验的一致性。

[0033] 因此,可以明白,可以用允许在一组设备之间部署计算环境(包括应用程序集合)的方式来有利地表示计算机系统。在一种这样的技术中,计算环境-包括应用程序集合、应用程序资源、以及由此使用的数据对象-是用以下方式表示的:它可被传递到各设备以供根据该设备的能力来呈现。该表示包括根据对象分层结构来组织并根据共同语法来表示的一组对象。这些对象包括计算机系统的数据对象,如用户创建的用户文件和数据。这些对象还包括构成操作系统组件的可执行二进制码和类库,如外壳以及其中提供的应用程序。该对象还包括指定计算环境的用户界面的信息,包括外壳偏好(例如,视觉主题、应用程序启动菜单、以及双击阈值)、用户帐户及特权、安全信息(例如,口令、安全权标、以及证书)、应用程序二进制码和配置信息、用户数据和元数据(例如,文件共享信息)、以及数据高速缓存(例如,最近使用的文件列表和浏览器历史)。尽管这些对象具有各种性质和用途,这些对象在对象分层结构中是以共同的方式来表示的,并且可在该分层结构中任意组织。因此,与包括一组隔离的数据存储(每一隔离存储包含一种类型的对象,例如,文件系统包含文件、注册表包含配置信息、以及数据高速缓存包含浏览器历史)的常规计算机环境相反,该方法中的对象分层结构以共同的方式将所有这些对象组织在该对象分层结构中。

[0034] 以此方式表示的计算环境可被传递到任何设备并以适合于该设备的能力的方式

来呈现。例如,工作站可以将该信息呈现为稳健且通用的计算环境,而公共工作站可以通过 web 浏览器呈现不同的计算环境体验(例如,作为可在用户的会话结束时丢弃的虚拟机),并且蜂窝电话可以提供具有对蜂窝电话相关信息(例如,联系人、日历、以及导航数据)的较快访问的较瘦界面。此外,对该信息集的更新(例如,偏好改变和对其中包含的数据文件的更新)可被应用于该信息集的规范源,并且因此被传播到该信息集被传递到的所有其他设备。同样,共享该计算环境的设备可以通过该共享信息集来集成,使得一个设备可以与在该信息集中标识的其他设备进行交互;例如,存储在第一设备上的数据可由第二设备访问,并且第一设备可由第二设备通过通信管道来控制。该信息集因此可以标识共享该计算环境的设备的集合连同每一设备的角色、能力、以及资源,以跨可能大量各种设备来提供综合计算体验。

[0035] 图 2 示出一个这样的场景,其中计算环境可由计算环境主机 42 主存,计算环境主机 42 可以存储并管理对象分层结构 44。计算环境主机 42 还可以用不同的方式代表各种设备(如蜂窝电话设备 46、个人笔记本计算机 50、以及公共工作站 54)并且还代表具有不同访问特权的不同类型的用户来呈现对象分层结构 44。该计算环境的呈现因此反映了跨展示相同的应用程序、用户简档、外壳配置、用户数据对象等的所有设备的一致计算环境。因此,用户可以通过高性能笔记本计算机访问该计算环境的全特征版本 52、在低功率蜂窝电话设备 46 上访问该计算环境的精简版本 48、以及通过公共终端 54 的 web 浏览器 46 访问该计算环境的可兼容浏览器并且面向隐私的版本 58。就每一这样的设备的能力支持该计算环境的呈现而言,由于根据每一设备的能力调整了对象分层结构 44 的呈现,所以可以呈现一致的用户界面和数据集。对计算环境的更新可以传播回计算环境主机 42,并且可以与其他设备自动进行同步。各设备还可以通过与其他设备共享存储在本地的数据并且通过控制其他设备或由其他设备控制来进行协作。因此,可将该计算环境设计并呈现为云计算体系结构,包括表达为跨形成同一计算环境的各协作门户(具有设备专用属性)的网格的所有设备(“客户机”)的一致呈现的设备无关表示(“云”)。具体而言,相对于该计算环境的应用程序,对象分层结构 44 可包括应用程序集合、应用程序资源、以及由此创建的数据对象的表示。此外,计算环境主机 42 可以将服务应用于构成对象分层结构 44 的各个对象,并且将对象存储在该对象分层结构中的共同格式可以便于这些服务的一致可用性和应用,而不论向其应用这些服务的对象的性质如何。

[0036] 除这些目的之外,对象分层结构方案可被设计成以特定方式来组织该对象分层结构的各个对象。例如,可以设计并构造各个对象以表示数据对象(如,文件)、用户和用户帐户、设备和该计算环境中发生的事件、以及来历、所有权、以及它们之间的相互关系。相对于对象的其他组织而言,组织对象的这一方式可以产生特定优点和/或减少缺点。因此,根据其来组织对象的该对象分层结构方案是鉴于这些对象和计算环境的使用来设计的。例如,可以通过将相应对象与具有各种许可集合的一个或多个用户帐户(例如,第一用户帐户具有对一对象的读/写许可,第二用户帐户具有对该对象的只读许可,并且第三用户帐户没有对该对象的许可)进行关联来对安全许可进行建模。计算环境可被配置成根据对象分层结构方案的结构和语义来利用该对象分层结构的对象,从而允许以适用于每一设备的配置和能力但在其他方面在所有这样的设备之间一致的方式来跨多个设备呈现可部署计算环境。

[0037] 在这一性质的对象分层结构内,一组应用程序可以用与每一应用程序的应用程序生存周期相一致的方式来管理。例如,计算环境主机 42 可以便于获取应用程序、将该应用程序与一个或多个设备相关联、以及将该应用程序部署在每一设备上。将应用程序表示成对象分层结构 44 中的一组对象可便于常见的应用程序管理任务。作为第一示例,并且在许多自动部署的场景中,应用程序可以容易地部署在该对象分层结构 44 中表示的多个计算机系统和设备上。例如,该应用程序的第一版本可以部署到个人笔记本计算机 50;该应用程序的第二版本可以部署成用于公共工作站 54 的 web 应用程序或启用 web 的应用程序;并且该应用程序的第三版本可以部署成供在低功率蜂窝电话设备 46 上使用的减少功能的应用程序。作为第二示例,计算环境主机 42 可以便于通过相应地调整对象分层结构 44 并通过将所主存和部署的该对象分层结构 44 的各版本进行同步来将这些调整部署到各设备以更新应用程序(例如,使用更新版本对应用程序资源进行修补或部署供与该应用程序一起使用的新内容)。

[0038] 将应用程序集合表示成可部署对象分层结构 44 中的一组对象还可启用新特征。作为第一示例,通过将对象分层结构 44 的特定部分部署到第二用户的设备、计算机系统、或计算环境主机 42,该计算环境可使用户能够与第二用户共享应用程序或应用程序会话的一方面。例如,第一用户可以选择一组歌曲对象来在第一计算机上通过媒体播放器应用程序进行播放,并可选择与第二用户共享该媒体播放器应用程序会话(例如,通过将媒体播放器应用程序的音频流传输到第二用户)。作为第二示例,应用程序的配置可以跨各设备和计算机系统来作为对象分层结构的一部分进行同步。设备无关应用程序属性(例如,添加到拼写检查字典的自定义单词)可以自动地传播到所有这样的设备和计算机系统,而计算环境主机 42 可为相应设备指定设备专用应用程序属性。以此方式,计算环境主机 42 的应用程序管理服务可以支持应用程序在由一个或多个用户所操作的各计算机系统和设备上的发现、部署、使用和移除。这些任务的许多部分可以通过调整对象分层结构 44 并将其部署同步到其中表示的各个设备来执行,作为自动化过程,这可以提供计算环境和总体用户体验的一致性的显著改进。

[0039] 图 3 示出这些技术的一个实施例,作为对将存储在应用程序存储中的应用程序部署到表示一设备上的计算环境的对象分层结构进行管理的示例性方法 60。参考图 2,这一示例性方法 60 可被用来通过将应用程序表示在(至少部分地)分发给一个或多个设备的可部署对象分层结构 44 中来在这些设备上安装并管理该应用程序,这些设备诸如蜂窝电话设备 40、笔记本计算机 50、以及公共工作站 54。示例性方法 60 在 62 开始,并且涉及接收将应用程序部署到设备的请求 64。该请求可以由希望安装该应用程序的计算环境用户通过运行在该计算环境在一设备上的表示中的程序生成,通过被配置成生成具有至少一个预先安装的应用程序的新计算环境的程序生成,等等。示例性方法 60 还涉及从应用程序存储获取该应用程序 66。如果该应用程序是 web 应用程序,则获取 66 可以仅涉及(例如)验证该应用程序在所提供的 URL 处的可用性。示例性方法 60 还涉及将对象分层结构中表示的设备映射到应用程序 68。映射 68 可涉及将设备在对象分层结构内的表示与应用程序在对象分层结构内的表示进行关联,在对象分层结构中相对于该设备为该应用程序创建一组配置信息,等等。

[0040] 图 3 的示例性方法还涉及在对象分层结构中部署应用程序 70。部署 70 可涉及在

对象分层结构内创建应用程序的表示,在对象分层结构的应用程序表示中表示一个或多个可执行二进制码,相对于该设备进一步配置该应用程序,等等。对于 web 应用程序,部署 70 可包括例如将到一网站的链接插入与该设备相关联的应用程序启动程序表示中,和 / 或在该设备上创建一个或多个 cookie 以便于在会话之间保存该 web 应用程序。示例性方法 60 还涉及将对象分层结构的至少一部分发送到设备 72。例如,诸如笔记本电脑 50 等高性能计算机系统可以接收对象分层结构的完全版本,而相对低功率的蜂窝电话设备 46 可以只接收对象分层结构中与所述蜂窝电话设备 46 相关的一部分,包括与蜂窝电话设备 16 相兼容的新应用程序的部署。在配置了对象分层结构以将应用程序与设备进行关联并将该对象分层结构部署到该设备之后,示例性方法 60 因此实现了对相对于该设备的所部署的应用程序的管理,并且因此在 74 结束。

[0041] 图 4 表示这些技术的第二实施例,被实现成用于对将存储在应用程序存储 84 中的应用程序部署到表示至少一个设备上的计算环境的对象分层结构 82 进行管理的示例性系统 86。这一场景 80 涉及三个可部署应用程序:计算器应用程序、日历应用程序、以及媒体播放器应用程序。可以使这些应用程序在三个设备上表示的计算环境内可用:蜂窝电话设备、笔记本电脑设备、以及公共工作站。在这一场景 80 中,应用程序由表示至少一个应用程序的应用程序存储 86 来主存。这些应用程序可以部署在计算环境内和设备上,通过重新配置对象分层结构 82 可以将该计算环境呈现在该设备上。

[0042] 在这一场景 80 中,应用程序管理系统 86 可用于实现一个或多个应用程序在对象分层结构内的部署。示例性系统 86 包括应用程序获取组件 88,该组件被配置成接收安装应用程序的请求并从应用程序存储 84 获取该应用程序。应用程序获取组件 88 还可确定该请求是否满足任何先决条件(例如,该应用程序是否可以在所请求的设备上运行;鉴于用户的各种安全特权和许可问题,用户是否被准许安装并运行该应用程序;等等)。图 4 的示例性系统 86 还包括应用程序映射组件 90,该组件被配置成将对象分层结构 82 中表示的至少一个设备映射到至少一个已安装应用程序。例如,如果计算器应用程序已经包括在对象分层结构 82 中,则应用程序映射组件 80 可以在对象分层结构 82 内创建该计算器应用程序与在其上提供该应用程序的一个或多个设备之间的关联。如果计算器应用程序尚未包括在对象分层结构 82 中,则示例性系统 86 还可在执行映射之前在对象分层结构 82 中创建该计算器应用程序的表示,例如通过将可执行二进制码的表示或到 web 应用程序网站的链接复制到对象分层结构 82。示例性系统 86 还包括应用程序部署组件 92,该组件被配置成将至少一个已安装应用程序部署在对象分层结构 82 中。部署可涉及例如,在对象分层结构内为新映射的设备创建设备专用配置信息;将新映射的设备与设备无关配置信息进行关联;将该应用程序插入应用程序启动程序以在一个或多个设备上呈现;等等。在将设备在对象分层结构 82 中的表示与该设备所呈现的计算环境中表示的应用程序集的表示进行关联之后,示例性系统 86 从而实现对表示该计算环境的对象分层结构中的部署应用程序的管理。

[0043] 本文所描述的技术可用许多方面的变型来实现,并且相对于这些和其他技术的其他变型,一些变型可呈现出附加优点和 / 或减少缺点。这些变型可以包括在各实施例中,如图 3 的示例性方法 60 和图 4 的示例性系统 86,并可以与其他这样的变型兼容以呈现若干这样的附加优点和 / 或减少缺点。因此,本领域普通技术人员可根据本文讨论的技术设计出许多这样的实施例。

[0044] 可在这些技术的各实现之间变化的第一方面涉及通过其提供应用程序管理服务的应用程序管理服务器的体系结构。如图 4 所示,应用程序管理系统 84 可以与具有各种应用程序的应用程序存储 82 交互,可以请求并接收其中存储的应用程序,并可以将该应用程序部署到对象分层结构 82。在这一场景 80 中,应用程序管理系统 84 被实现成一个或多个应用程序存储与计算环境主机 86 之间的独立接口。然而,其他体系结构和系统因素也可适用于实现这些技术。

[0045] 图 5 示出第二示例性体系结构 100,其中应用程序管理服务 86 还用作应用程序存储 84,并且因此包含可部署到对象分层结构 82 的各种应用程序,如计算器应用程序 102 和日历应用程序 104。这一实施例可呈现应用程序获取服务与可用于部署的应用程序之间较紧密集成的优点。

[0046] 图 6 示出第三示例性体系结构 110,其中计算环境主机 42 包括应用程序管理服务 86。因此,计算环境主机 42 可以存储对象分层结构 82 和应用程序管理服务 86 的各个服务,如应用程序获取组件 88、应用程序映射组件 90、以及应用程序部署组件 92。在该第三示例性体系结构 110 中,应用程序 112 可以从设备直接接收,如通过用户在该设备上表示的计算环境内发起的请求。计算环境主机 42 可以接收这一请求并通过应用程序管理服务 86 的各组件来处理它,这将更新存储在计算环境主机 42 中的对象分层结构 82。经更新的对象分层结构 82 随后可以与该设备进行同步以完成应用程序到该设备的部署。在实现本文讨论的技术时,本领域普通技术人员可以设计出这些和其他系统因素。

[0047] 可在这些技术的各实现之间变化的第二方面涉及应用程序管理的基本组件(包括获取、映射以及部署)中的变型。作为这一方面的第一变型,获取应用程序通常需要接收部署该应用程序的请求以及获取该应用程序来相对于特定设备部署到对象分层结构。然而,获取还可包括通过该设备向用户呈现应用程序许可证,如显示该应用程序的最终用户许可协议的条款和条件。获取还可包括请求该用户接受应用程序许可证来作为部署的先决条件,以及在接收到对应用程序许可证的接受后,获取可包括记录对该应用程序许可证的接受,如通过通知应用程序许可证服务器。这一接受可以作为来自用户的确认或作为自动接受来接收,例如在用户预先授权接受特定应用程序许可证下的任何应用程序的情况下(例如,在 GNU 公共许可证的特定版本下提供的任何应用程序)。获取还可包括对与至少一个设备的用户的获取事务进行中介,如对该应用程序的许可证的电子商务购买。获取还可包括生成表示应用程序的用户的用户帐户,这对获取对 web 应用程序的访问而言尤其有利,如整个地主存在 web 邮件服务器上并通过计算环境中的浏览器来提供的 web 邮件应用程序。

[0048] 作为这一第二方面的第二变型,应用程序的映射通常涉及将该应用程序与对象分层结构中表示的一个或多个设备进行关联。然而,该映射还可包括检查设备专用条件,如该应用程序是否能以与该设备相兼容的版本可用以及该应用程序或其变型(如,较早版本)是否已经在该设备上可用。映射还可涉及获取附加资源以促进应用程序到新设备的映射,如可以与该设备兼容的应用程序二进制码的新版本或该应用程序所依赖的可被部署到该设备的附加组件(例如,可以从另一服务器检索的支持运行时)。映射还可涉及通过应用程序激活服务器来激活应用程序,这可以授权该应用程序在特定设备上执行,如通过通过可由该设备上的应用程序在启动时验证的认证证书。

[0049] 作为这一第二方面的第三变型,应用程序的部署涉及将该应用程序部署在对象分层结构内,如通过将该应用程序的各组件的表示插入该对象分层结构中。可以用多种方式来构造更通常组织的应用程序。小实用程序可包括可以全部或部分地编译的单个可执行二进制码,或包括可以作为在每一设备本地编译的源代码来包括的单个可执行二进制码。这一可执行二进制码可以表示成例如源代码模块;解析树;表示部分编译的二进制码、全部编译的二进制码的中间语言指令,等等。较大实用程序可包括一组资源,如一个或多个可执行二进制码;可在全局汇编高速缓存中注册的共享汇编;诸如图标、声音对象、图片、文档、以及视频等应用程序资源;帮助文件;存储在诸如关系数据库等一个或多个结构化数据存储中的数据资源;辅助应用程序资源,如用于文字处理应用程序中的拼写检查工具的语言字典文件;用作其中指定的资源的索引的应用程序清单;等等。此外,应用程序可包括可执行二进制码的若干版本,其中的每一个版本都可针对特定平台;例如,应用程序可包括供在诸如笔记本电脑 50 等较强大设备上使用的全特征版本和供在低功率蜂窝电话设备 46 上使用的减少功能版本。应用程序还可包括若干组配置信息,每一组信息都针对特定平台。应用程序可被呈现为独立安装程序包,或可由基于设备的安装程序部署的可安装档案,或可在任何位置利用的一组一个或多个资源,等等。

[0050] 对象分层结构还可包括较不常规结构的应用程序。例如,游戏应用程序可包括大量图形资源,如用于构造复杂三维环境和其中表示的对象的位图纹理和网格。启用 web 的应用程序可包括一组网页,它们可包含内联代码的各部分,或可以引用外部代码资源,如以源代码或已编译形式包括的 Java 对象。这样的应用程序还可以简单地被表示成对其中可获得这样的资源的网站的引用。还可以通过诸如可扩展应用程序标记语言 (XAML) 等标记语言来构造启用 web 的应用程序,它可在设备上作为应用程序或 web 浏览器中主存的小应用程序来编译或解释性地执行。最后,web 应用程序可以简单地表示成引用在其上部署该应用程序的部分或全部的一个或多个网站的一个或多个 URL,它可以通过 web 浏览器在该设备上访问;作为替换或补充,可包括一些资源以用于设备部署,如可存储在本地以持久保持 web 应用程序在会话之间的状态的 cookie。

[0051] 因为构成应用程序的资源可以是不同的,所以应用程序在对象分层结构中的部署可以类似地稳健。例如,文本资源可以作为纯文本来包括,或可以使用通用或专用压缩实用程序来压缩。二进制资源可以被呈现成串行化二进制码,可以用二进制压缩实用程序来压缩,可被进行代码转换(例如,经由 Uuencode 或 MIME),等等。可以根据诸如 XML 等分层格式化来构造分层组织的对象,或可以将其串行化成单个分层对象。大型对象可以作为原子单元来包括、可被压缩、被进行代码转换(例如,电影文件可被以较低比特率重新编码以节省空间)、被分割成可随时间或在自组织的基础上被单独地发送到设备的各个部分,等等。资源可被包括在对象分层结构中,或可以作为对主存可按需访问的资源的服务器的引用来插入。多个对象的大型集合可以作为档案来包括(例如,包含数千对象的压缩文件)。基于 web 的应用程序可被表示成其中可获得该应用程序的各组件的一个或多个 URL;作为替换或补充,可以捕捉该 web 应用程序的快照并将其包括在对象分层结构中以提高该应用程序的高速缓存版本或启用对该应用程序的离线访问。在实现本文讨论的技术时,本领域普通技术人员可能能够设计出用于在对象分层结构中表示应用程序的许多技术。

[0052] 除在对象分层结构中表示应用程序之外,部署还可包括相对于设备和 / 或相对于

用户来配置该应用程序。例如,可为设备制定新配置信息集合,或可以指定配置信息集合库中的适当的配置信息集合以供该设备使用。图 7 示出一个这样的场景 120,其中对象分层结构 82 包括包含两个可执行二进制码(用于本地本机执行的一个已编译可执行码,以及用于通过标准 web 浏览器进行较慢但更标准化的访问的一个客户机侧浏览器脚本)和两个配置集合(用于高性能设备的全功能配置集合和用于低功率移动设备的便携式配置集合)的日历应用程序的表示。对象分层结构 82 还包括包含供与任何设备一起使用的设备无关设置的配置集合,如用户所选择的首选色彩方案。对象分层结构 82 还包含可在其上呈现该计算环境的两个设备的表示:低功率蜂窝电话设备 46 和高性能笔记本电脑 50。

[0053] 在对象分层结构 82 内为这些设备部署应用程序时,应用程序部署组件 92 可以针对在每一设备上的特定使用来配置该应用程序。例如,可以通过将日历应用程序与可以在蜂窝电话设备 46 的 web 浏览器应用程序内运行的客户机侧浏览器脚本可执行二进制码进行关联并与用于低功率设备的便携式配置集合进行关联来为蜂窝电话设备 46 部署该应用程序。类似地,可以通过将日历应用程序与已编译可执行二进制码进行关联并与可提供笔记本电脑 50 能够呈现的计算上密集的特征的全功能配置集合进行关联来为笔记本电脑 50 部署该应用程序。然而,两设备可以合并设备无关配置集合以呈现一致的设备无关特征集合,从而促进计算网格的所有这些设备之间的一致计算环境。在实现本文讨论的技术时,本领域普通技术人员能够设计出在对象分层结构 82 中对应用程序的获取、映射、以及部署的附加变型。

[0054] 可在这些技术的实现之间变化的第三方面涉及可被添加到这些实施例的基本组件的附加特征。除基本管理服务(获取、映射、以及部署)之外,可为应用程序管理提供附加服务,这可便于用户和计算机系统管理应用程序生存周期。

[0055] 作为这一第三方面的第一变型,应用程序管理服务还可担当计算环境主机,如图 6 的第三示例性体系结构 110 中示出的。在这一第一变型中,应用程序管理服务还包括被配置成存储对象分层结构的对象分层结构存储以及被配置成将对象分层结构的至少一部分发送到这些设备(例如,通过将经更新的对象分层结构与存储在每一设备上的对象分层结构进行同步)的对象分层结构传输组件。对象分层结构的包括已部署到该设备的应用程序的那一部分的传输可以实现应用程序在该设备上的安装和部署。

[0056] 作为这一第三方面的第二变型,应用程序管理服务还可便于发现用户可能感兴趣的程序。例如,对应用程序到对象分层结构的部署进行管理的方法(如图 3 的示例性方法 60)可包括查询方面,其中该方法被配置成在从接收到应用程序查询后选择满足该应用程序查询的应用程序并将所选应用程序呈现给该用户。类似地,用于对应用程序到对象分层结构的部署进行管理的系统(如图 4 的示例性系统 86)可包括应用程序发现组件,该组件可被配置成从用户接收应用程序查询,选择满足该应用程序查询的应用程序,并将所选应用程序呈现给用户。用户随后可以选择获取所选应用程序中的一个或多个,应用程序管理系统可将它们部署到对象分层结构。

[0057] 图 8 示出一个这样的场景 130,其中用户 132 希望标识可与用户的蜂窝电话设备 46 兼容的音频播放器应用程序。用户 132 可以因此向应用程序管理服务 86 的应用程序发现组件 134 发起应用程序查询 136,如对蜂窝电话设备兼容的媒体播放器应用程序的查询。应用程序发现组件 134 可以与应用程序存储 84 进行通信以标识满足应用程序查询 136 的

应用程序,如用于蜂窝电话设备的音频播放器应用程序和用于蜂窝电话设备的视频播放器应用程序。应用程序发现组件 134 可以因此提供将这些应用程序呈现给用户 132 的响应 138。用户 132 可以使用对所发现的要部署到蜂窝电话设备 46 的应用程序中的一个或多个的选择 140 来回复应用程序管理服务 86。应用程序管理服务 86 随后可以获取所选应用程序(在这一场景 130 中,是用于蜂窝电话设备的音频播放器应用程序),并可以将该应用程序部署到对象分层结构 142,它随后可被发送到蜂窝电话设备 46 以完成所选应用程序的部署。在实现本文讨论的技术时,本领域普通技术人员能够设计出用于便于应用程序的发现的其他方法。

[0058] 作为这一第三方面的第三变型,应用程序管理服务还可便于启动应用程序。例如,实现了应用程序管理服务的示例性系统可以包括应用程序启动组件,该组件可被配置成在至少一个设备上启动应用程序。应用程序启动组件可包括例如被配置成为该设备提供平台并便于应用程序在其上的执行的运行时。例如,应用程序可以作为客户机侧浏览器脚本部署在服务中,并且应用程序管理服务可包括对象分层结构连同该浏览器脚本可在其中解释性地执行的浏览器运行时。作为另一示例,应用程序可被表示成源代码,并且应用程序启动组件可包括在设备上启动应用程序后将该源代码编译成可在该设备上执行的可执行二进制码的设备专用编译器。此外,应用程序启动组件还可存储已编译的可执行二进制码,如存储在汇编高速缓存中,并可通过读取所存储的可执行二进制码而非重新编译该应用程序来对启动该应用程序的后续请求作出响应。作为第三示例,启用 web 的应用程序可包括被配置成访问远程资源的客户机侧前端,并且在启动该应用程序时,应用程序启动组件可以尝试连接到该远程资源并验证其对设备的可用性。在实现本文讨论的技术时,本领域普通技术人员能够设计出便于在设备上启动应用程序的许多方式。

[0059] 作为这一第三方面的第四变型,应用程序管理服务还可便于与其他用户共享应用程序的各方面。这样的应用程序共享可以在许多场景中发生。作为第一示例,该共享可涉及共享全应用程序(包括与其相关联的所有资源和数据对象)。作为第二示例,该共享可涉及与另一用户共享该应用程序的一个版本;例如,第一用户可能购买了用于使用该应用程序的许可证,并可与第二用户共享该应用程序的减少功能版本,这可展示该应用程序的一些特征以鼓励第二用户取得该应用程序的许可证。作为第三示例,该共享可涉及共享与该应用程序相关联的一个或多个数据对象,如使用该应用程序创建的媒体对象。第二用户随后可以使用同一应用程序或另一应用程序来在该第二用户的计算环境内呈现这些媒体对象。作为第四示例,该共享可涉及与另一用户共享应用程序的配置;例如,在作为团队来开发软件时,软件开发团队的若干成员可以利用集成开发环境的特定配置。作为第五示例,该共享可涉及与另一用户共享应用程序的会话;例如,两个用户可以在准备文本文档时进行协作,并且第一用户可以与第二用户共享文字处理应用程序的会话以促进这些用户之间的实况交互。这些共享场景还可以在(由单个对象分层结构表示的)计算环境的若干用户之间产生,或在(由多个对象分层结构表示的)两个或多个计算环境的用户之间产生,或可以在通过经由诸如文件传输协议(FTP)等常见通信协议来提供共享计算环境的数据对象来与以未知方式表示的计算环境的用户共享表示一计算环境的对象分层结构的一部分时产生。在实现本文讨论的技术时,本领域普通技术人员设计出这些和其他共享场景。

[0060] 图 9 示出应用程序共享场景 150,其中第一对象分层结构 152 的第一用户获取了

媒体播放器应用程序和各艺术家创建的各种歌曲的形式的一些媒体对象,其中这些媒体对象是以高保真度方式来编码的(即,以 320kbps 编码速率)。第一用户还创建诸如播客等媒体播放器会话,它引用在第一对象分层结构 152 的媒体部分中表示的对象中的几个。第一用户可能希望与第二对象分层结构 154 的第二用户共享这一播客。应用程序共享因此可以涉及将媒体播放器会话从第一对象分层结构 152 复制到第二对象分层结构 154。因为媒体播放器会话引用存储在第一对象分层结构 152 中的一些媒体对象,所以应用程序共享还可涉及通过将其中引用的媒体对象复制到第二对象分层结构 154 来共享这些媒体对象。然而,与媒体对象相关联的数字权限可以只准许以较低保真度来将媒体对象传输到其他对象分层结构;因此,计算环境可以将经代码转换成较低质量(24kbps 编码速率)的媒体对象的版本发送到第二对象分层结构。另外,第二对象分层结构 154 可能不包括用于由媒体对象表示的类型的音频的媒体播放器应用程序。因此,第一对象分层结构 152 可以与第二对象分层结构 154 共享媒体播放器应用程序的某一版本;然而,可遵循第一用户所获取的媒体播放器应用程序的最终用户许可协议来将减少功能版本发送到第二用户。以此方式,该应用程序、由此呈现媒体对象和第一用户所生成的应用程序会话可通过将第一对象分层结构 152 的各部分发送到第二对象分层结构 154 来以自动化方式与第二用户共享,同时遵循与该应用程序会话相关的各组件的数字权限。

[0061] 作为这一第三方面的第五变型,应用程序管理服务可以便于用新内容更新应用程序。通常通过修补二进制码、插入新资源、以及更改或移除现有资源等来使用新内容将应用程序更新成新版本并用新内容来更新应用程序。这样的更新可由应用程序管理服务通过获取应用程序更新并将这些更新部署到对象分层结构中表示的应用程序来执行(例如,通过更新对象分层结构中包含的应用程序资源和/或将新资源作为新对象插入该对象分层结构)。这些更新可以在获取了要由应用程序展示的新内容的用户进行请求时应用;或可由应用程序管理服务通过向应用程序存储进行查询来检索;或可由应用程序存储或管理服务推送到应用程序管理服务;等等。经更新的对象分层结构随后可被发送到一个或多个设备,并且因此可以在减少了对用户参与的依赖的情况下实现对计算环境的应用程序的更新。在实现本文讨论的技术时,本领域普通技术人员能够设计出更新对象分层结构中表示的应用程序的许多方式。

[0062] 作为这一第三方面的第六变型,应用程序管理服务还可准许从计算环境中撤消应用程序。这一撤消可涉及许多场景,如移除应用程序到设备的映射、在应用程序激活服务器的授权设备列表中解除设备的授权、将应用程序资源从设备移除(例如,通过从设备扣押对象分层结构中包含应用程序资源的那一部分)、以及通过从对象分层结构并从所有映射的设备移除所有应用程序资源来完全停止使用该应用程序。该撤消可以是在希望从一个或多个设备卸载该应用程序的用户进行请求时来进行的,或在有限持续时间的软件许可证期满时进行的,或在从计算网络移除设备时进行的,等等。作为这一变型的一示例性实施例,应用程序管理系统可包括被配置成以这些方式中的任一种从至少一个设备撤消应用程序的应用程序撤消组件。在实现本文讨论的技术时,本领域普通技术人员能够设计出从对象分层结构所表示的计算环境内撤消应用程序的许多方式。

[0063] 这些方面的这些变型可包括在本文讨论的技术的许多实施例中,包括图 3 的示例性方法 60 和图 4 的示例性系统 86。此外,若干这样的变型可以作为组合来实现,以及与这

些技术的这些和其他方面的其他变型一起实现,以相对于图 3-4 中示出的更基本实施例来提供若干优点和 / 或减少缺点。一些组合还可提供复合优点;例如,应用程序发现组件可以协作式地链接到应用程序获取组件以呈现应用程序的取得的发现、购买、许可证接受、以及用户帐户创建来作为综合用户体验。

[0064] 图 10 示出用于对应用程序在表示至少一个设备(如蜂窝电话设备 46、笔记本电脑 50、公共工作站 54)上的计算环境的对象分层结构 44 中的部署进行管理的示例性系统 160。图 10 的示例性系统 160 实现本文讨论的各变型中的许多变型。示例性系统 160 包括被配置成存储对象分层结构 44 的对象分层结构存储 82 和表示至少一个应用程序的应用程序存储 84,相应应用程序包括至少一个应用程序资源。示例性系统 160 还包括应用程序发现组件 134,该组件被配置成从用户接收应用程序查询,选择满足该应用程序查询的应用程序,并将所选应用程序呈现给用户。示例性系统 160 还包括应用程序获取组件 88,该组件被配置成接收安装应用程序的请求,通过该设备将应用程序许可证呈现给用户,以及在从用户接收到对该应用程序许可证的接受后记录对该应用程序许可证的接受。应用程序获取组件 88 还被配置成生成表示应用程序的用户的用户帐户以及对与至少一个设备的用户的获取事务进行中介。应用程序发现组件 134 与应用程序获取组件 88 一起便于典型的应用程序生存周期中的应用程序取得。

[0065] 图 10 的示例性系统 160 还包括应用程序映射组件 90,该组件被配置成将对象分层结构 44 中表示的至少一个设备映射到至少一个已安装应用程序。示例性系统 160 还包括应用程序部署组件 92,该组件被配置成通过相对于设备和该设备的用户中的至少一个来配置应用程序以将至少一个已安装应用程序部署在对象分层结构 44 中。应用程序映射组件 90 和应用程序部署组件 92 一起便于典型的应用程序生存周期中、在对象分层结构内的应用程序安装。

[0066] 图 10 的示例性系统 160 还包括应用程序启动组件 162,该组件被配置成在至少一个设备上启动应用程序。示例性系统 160 还包括应用程序共享组件 164,该组件被配置成接收与用户共享应用程序的请求,以及在接收到共享该应用程序的请求后与该用户共享对象分层结构中包括以下至少一个的一部分:应用程序、应用程序会话、应用程序配置、以及该应用程序生成的至少一个数据对象。示例性系统 160 还包括应用程序更新组件 166,该组件被配置成更新对象分层结构 44 中的至少一个应用程序资源。应用程序启动组件 162、应用程序共享组件 164、以及应用程序更新组件 166 一起便于典型的应用程序生存周期中、在一个或多个设备上的应用程序执行。

[0067] 图 10 的示例性系统 160 还包括应用程序撤消组件 168,该组件被配置成撤消应用程序(例如,通过从对象分层结构中的一个或多个设备的表示中解除该应用程序的映射、将应用程序资源从一个或多个设备移除、和 / 或通过将该应用程序从对象分层结构完全移除来完全停止使用它)。应用程序撤消组件 168 因此便于在典型的应用程序生存周期中的应用程序卸载。

[0068] 最后,示例性系统 160 还包括对象分层结构传输组件 170,该组件被配置成将对象分层结构 44 的至少一部分发送到至少一个设备。对象分层结构 44 可由应用程序管理系统 160 的各组件中的许多组件来修改,并且对对象分层结构 44 的修改可以与其中表示的一个或多个设备进行同步以实现计算环境的应用程序集合的更新。通过组合本文讨论的技术

的这些变型,示例性系统 160 因此提供用于在构成计算网络的许多设备之间经由典型的应用程序生存周期管理大型应用程序集合的全面的应用程序管理服务。

[0069] 尽管用结构特征和 / 或方法动作专用的语言描述了本主题,但可以理解,所附权利要求书中定义的主题不必限于上述具体特征或动作。相反,上文所描述的具体特征和动作是作为实现权利要求的示例形式来公开的。

[0070] 如在本申请中所使用的,术语“组件”、“模块”、“系统”、“接口”等一般旨在表示计算机相关的实体,该实体可以是硬件、硬件和软件的组合、软件、或者执行中的软件。例如,组件可以是,但不限于是,在处理器上运行的进程、处理器、对象、可执行码、执行的线程、程序和 / 或计算机。作为说明,运行在控制器上的应用程序和控制器都可以是组件。一个或多个组件可以驻留在进程和 / 或执行的线程中,并且组件可以位于一个计算机内和 / 或分布在两个或更多计算机之间。

[0071] 此外,所要求保护的主体可以使用产生控制计算机以实现所公开的主体的软件、固件、硬件或其任意组合的标准编程和 / 或工程技术而被实现为方法、装置或制品。在此使用的术语“制品”旨在涵盖可以从任何计算机可读设备、载体或介质访问的计算机程序。当然,本领域的技术人员将会认识到,在不背离所要求保护的主体范围或精神的前提下可以对这一配置进行许多修改。

[0072] 图 11 和以下讨论提供了对用于实现此处所阐述的原理中的一个或多个实施例的合适计算环境的简要、概括描述。图 11 的操作环境只是合适的操作环境的一个示例,并不旨在对该操作环境的使用范围或功能提出任何限制。示例计算设备包括但不限于个人计算机、服务器计算机、手持式或膝上型设备、移动设备(诸如移动电话、个人数字助理(PDA)、媒体播放器等等)、多处理器系统、消费电子产品、小型计算机、大型计算机、包括任何以上系统或设备的分布式计算环境等等。

[0073] 尽管并非必需,但各实施例在由一个或多个计算设备执行的“计算机可读指令”的一般上下文中描述。计算机可读指令可通过计算机可读介质来分发(在以下讨论)。计算机可读指令可被实现为执行特定任务或实现特定抽象数据类型的程序模块,如函数、对象、应用程序编程接口(API)、数据结构等等。通常,计算机可读指令的功能可按需在各个环境中组合或分布。

[0074] 图 11 示出了包括被配置成实现此处所提供的的一个或多个实施例的计算设备 182 的系统 180 的示例。在一种配置中,计算设备 182 包括至少一个处理单元 186 和存储器 188。取决于计算设备的确切配置和类型,存储器 188 可以是易失性的(如 RAM)、非易失性的(如 ROM、闪存等)或是两者的某种组合。该配置在图 11 中由虚线 184 来示出。

[0075] 在其他实施例中,设备 182 可以包括附加特征和 / 或功能。例如,设备 182 还可以包括附加存储(例如,可移动和 / 或不可移动),其中包括但不限于磁存储、光存储等等。这样的附加存储在图 11 中由存储 190 示出。在一个实施例中,用于实现此处所提供的的一个或多个实施例的计算机可读指令可以在存储 190 中。存储 190 还可以储存实现操作系统、应用程序等其他计算机可读指令。可以在存储器 188 中加载计算机可读指令以便由例如处理单元 186 来执行。

[0076] 此处所使用的术语“计算机可读介质”包括计算机存储介质。计算机存储介质包括以用于存储诸如计算机可读指令或其他数据等信息的任何方法或技术来实现的易失性

和非易失性、可移动和不可移动介质。存储器 188 和存储 190 都是计算机存储介质的示例。计算机存储介质包括但不限于, RAM、ROM、EEPROM、闪存或其他存储器技术、CD-ROM、数字多功能盘 (DVD) 或其他光存储、磁带盒、磁带、磁盘存储或其他磁存储设备、或者可用于存储所需信息并且可由设备 182 访问的任何其他介质。任何这样的计算机存储介质都可以是设备 182 的一部分。

[0077] 设备 182 还可包括允许该设备 182 与其他设备进行通信的通信连接 196。通信连接 196 可包括但不限于, 调制解调器、网络接口卡 (NIC)、集成网络接口、射频发射机 / 接收机、红外线端口、USB 连接、或用于将计算设备 182 连接到其他计算设备的其他接口。通信连接 196 可以包括有线连接或无线连接。通信连接 196 可以发送和 / 或接收通信媒体。

[0078] 术语“计算机可读介质”可以包括通信介质。通信介质通常以诸如载波或其他传输机制等“已调制数据信号”来体现计算机可读指令或其他数据, 并包括任何信息传送介质。术语“已调制数据信号”指以对信号中的信息进行编码的方式设置或改变其一个或多个特征的信号。

[0079] 设备 182 可包括输入设备 194, 诸如键盘、鼠标、笔、语音输入设备、触摸输入设备、红外照相机、视频输入设备和 / 或任何其他输入设备。设备 182 中还可包括输出设备 192, 诸如一个或多个显示器、扬声器、打印机和 / 或任何其他输出设备。输入设备 194 和输出设备 192 可以通过有线连接、无线连接或其任何组合来连接到设备 182。在一个实施例中, 来自另一计算设备的输入设备或输出设备可用作计算设备 182 的输入设备 194 或输出设备 192。

[0080] 计算设备 182 的组件可以通过诸如总线等各种互连来连接。这些互连可包括诸如 PCI Express 等外围部件互连 (PCI)、通用串行总线 (USB)、火线 (IEEE 1394)、光学总线结构等。在另一实施例中, 计算设备 182 的组件可以通过网络互连。例如, 存储器 188 可由位于通过网络互连的不同物理位置的多个物理存储器单元组成。

[0081] 本领域技术人员会认识到, 用于存储计算机可读指令的存储设备可分布在网络上。例如, 可以通过网络 198 访问的计算设备 200 可以存储实现此处所提供的一个或多个实施例的计算机可读指令。计算设备 182 可以访问计算设备 100 并下载部分或全部计算机可读指令来执行。或者, 计算设备 182 可以按需下载计算机可读指令的片断, 或者某些指令可以在计算设备 182 处执行而某些则在计算设备 200 处执行。

[0082] 此处提供了各实施例的各种操作。在一个实施例中, 所描述的操作中的一个或多个可以组成储存在一个或多个计算机可读介质上的计算机可读指令, 这些指令如果被计算设备执行则使得计算设备执行所描述的操作。描述一些或全部操作的顺序不应被解释为暗示了这些操作必须要依赖于顺序。从本说明书获益的本领域技术人员应理解替换的排序。此外, 应该理解并非所有的操作都必须存在于此处所提供的每一实施例中。

[0083] 此外, 在此使用词语“示例性”意指用作示例、实例或说明。在此被描述为“示例性”的任何方面或设计并不一定被解释为比其他方面或设计有利。相反, 使用词语“示例性”旨在以具体的方式呈现各个概念。如本申请中所使用的, 术语“或”意指包括性“或”而非互斥性“或”。即, 除非另有指定或从上下文可以清楚, 否则“X 使用 A 或 B”意指任何自然的包括性排列。即, 如果 X 使用 A ; X 使用 B ; 或 X 使用 A 和 B 两者, 则在任何以上情况下, 都满足“X 使用 A 或 B”。另外, 本申请中和所附权利要求书中所使用的冠词“一”和“一个”一般可

被解释为是指“一个或多个”，除非另有指定或从上下文可以清楚指的是单数形式。

[0084] 同样，虽然参考一个或多个实现示出并描述了本发明，但本领域技术人员基于对本说明书和附图的阅读和理解可以想到各种等效替换和修改。本发明包括所有这些修改和替换并且只由所附权利要求书来限定。特别地，对于由上述组件（例如，元素、资源等）执行的各种功能，除非另外指明，否则用于描述这些组件的术语旨在对应于执行所描述的执行此处在本发明的示例性实现中所示的功能的组件的指定功能（例如，功能上等效）的任何组件，即使这些组件在结构上不等效于所公开的结构。另外，尽管可相对于若干实现中的仅一个实现来公开本发明的一个特定特征，但是这一特征可以如对任何给定或特定应用所需且有利地与其它实现的一个或多个其它特征相组合。此外，就在说明书或权利要求书中使用术语“包含”、“具有”、“含有”和“带有”及其变体而言，这些术语旨在以与术语“包括”相似的方式为包含性的。

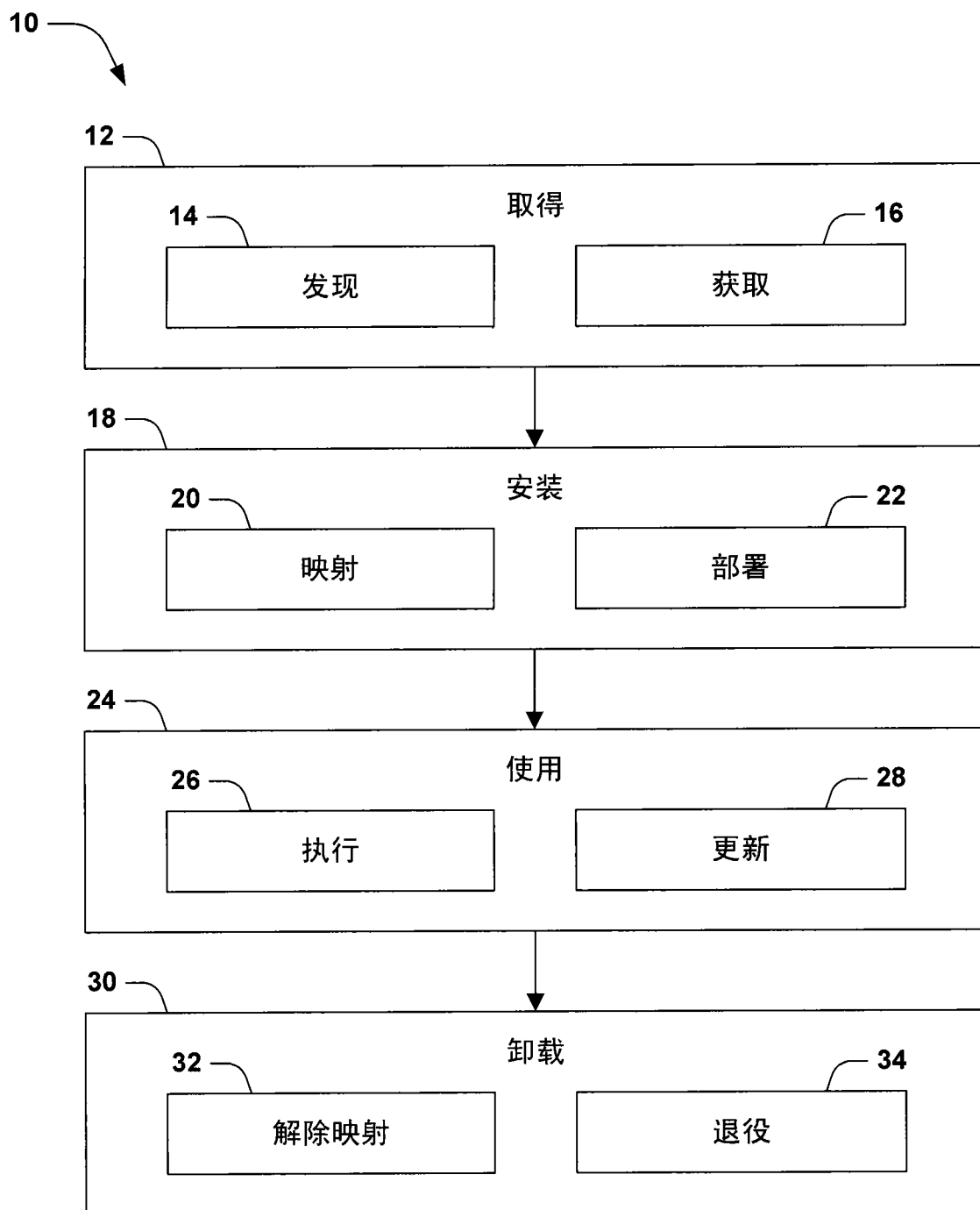


图 1

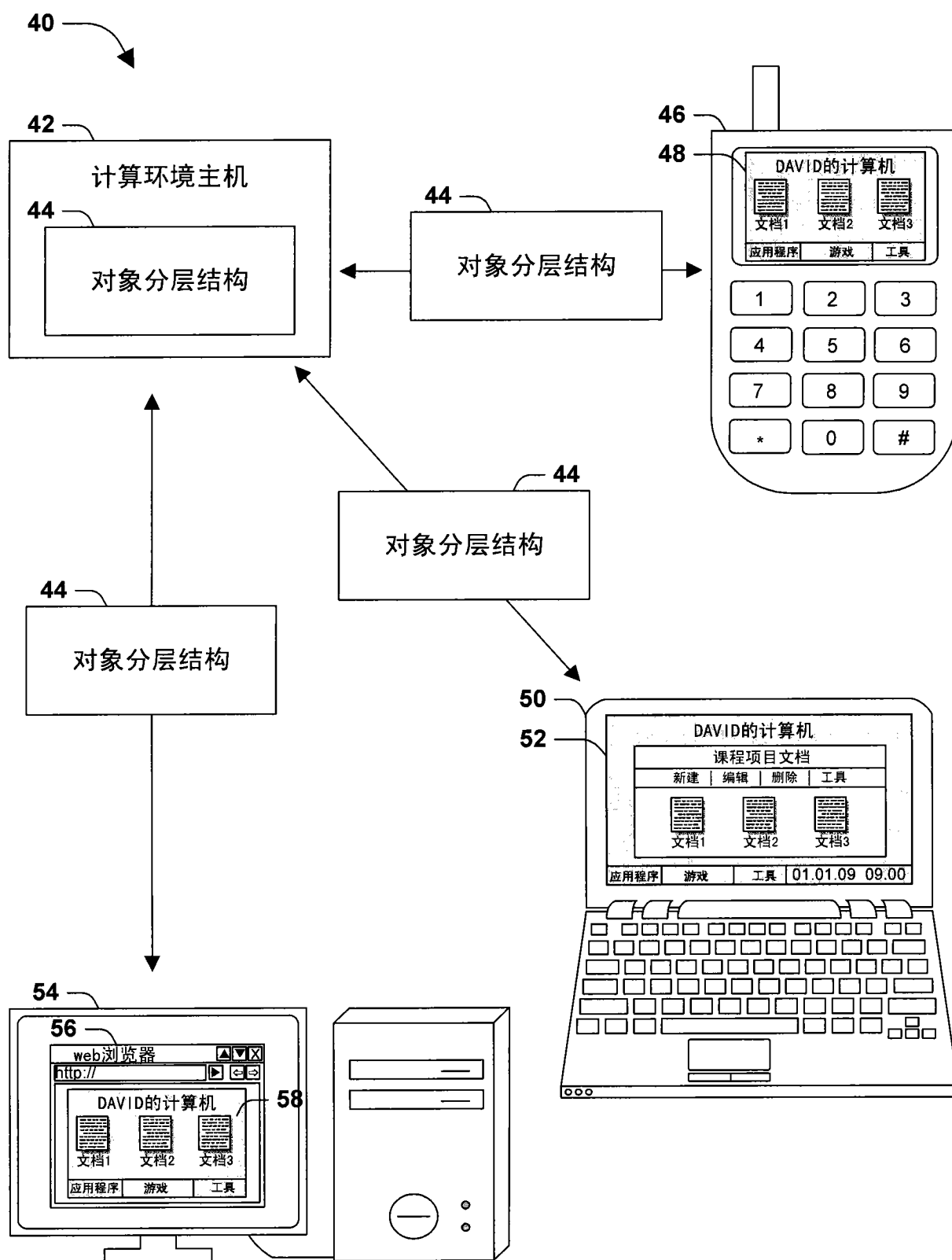


图 2

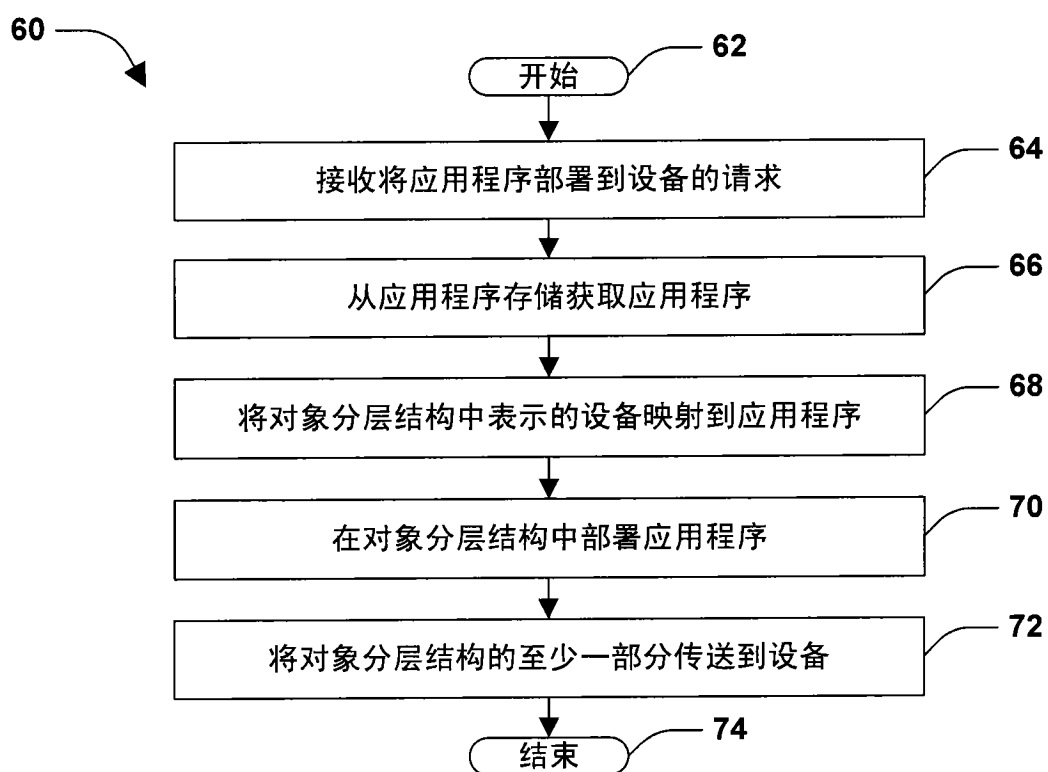


图 3

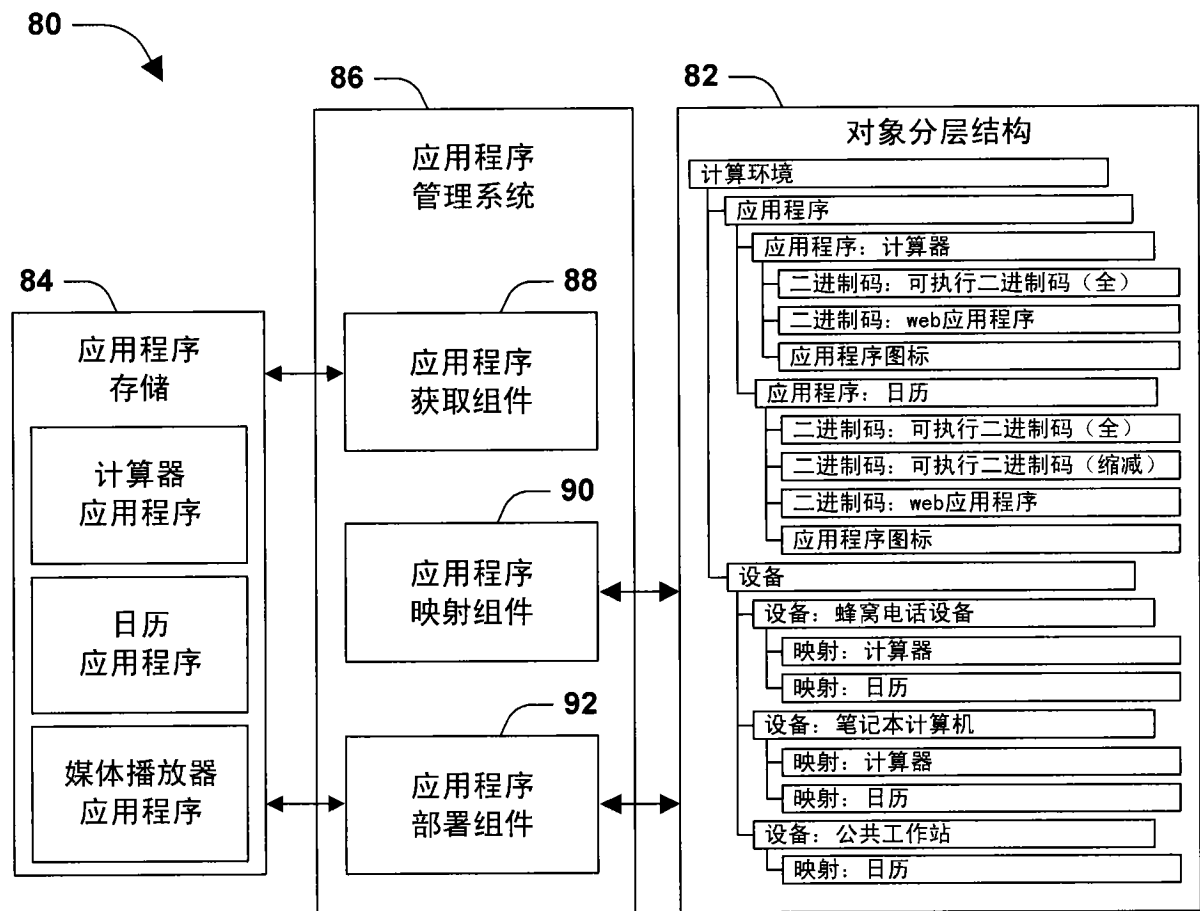


图 4

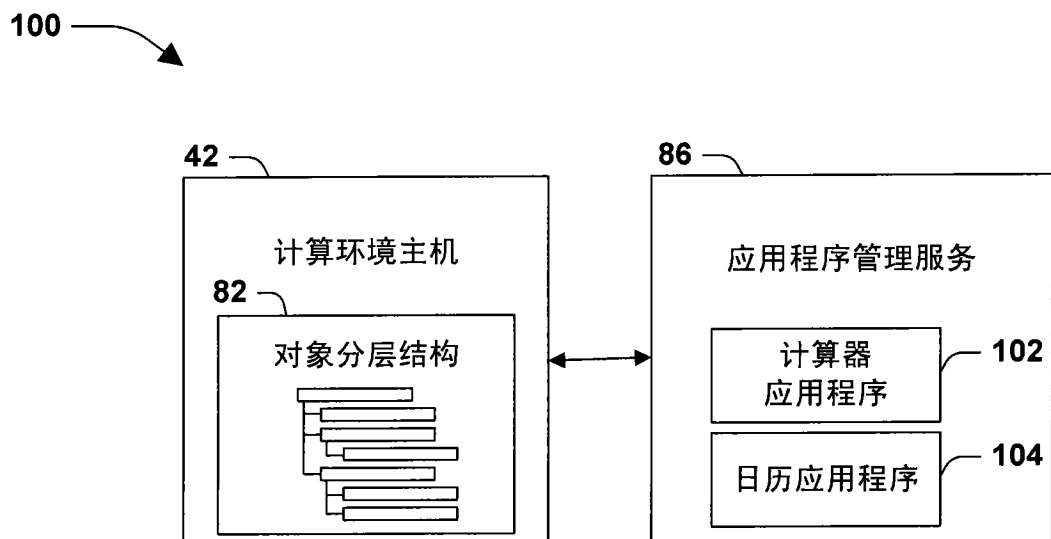


图 5

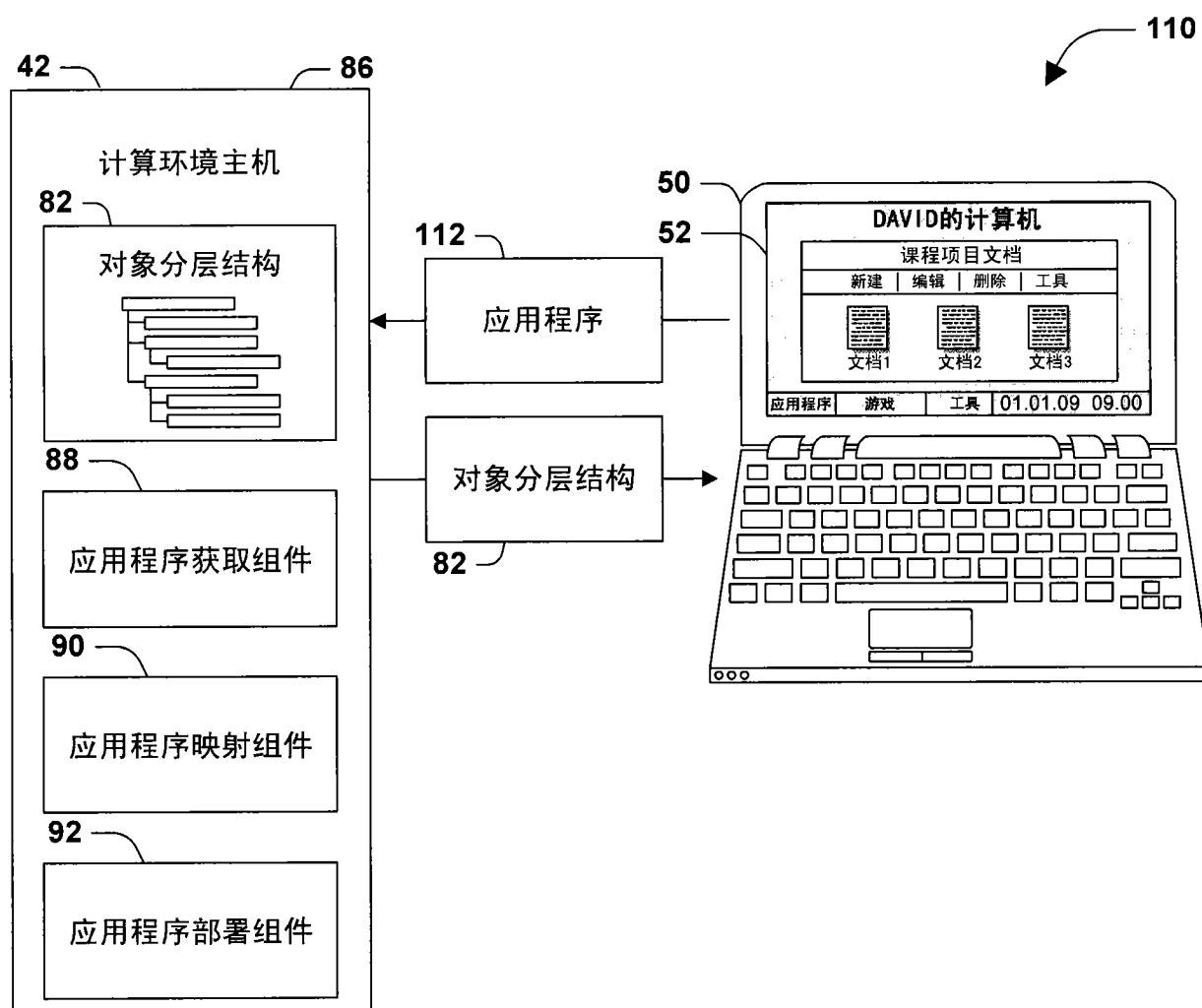


图 6

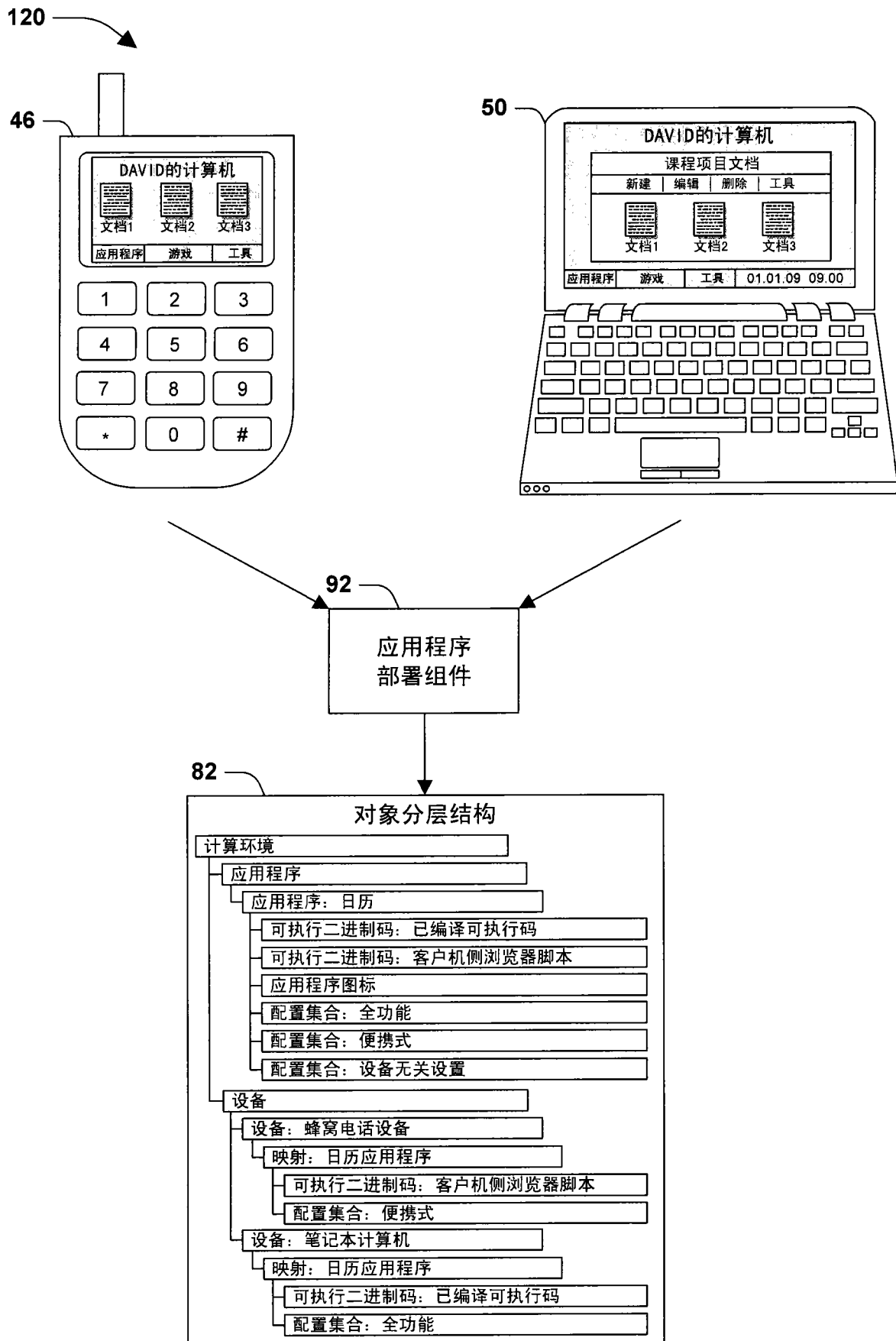


图 7

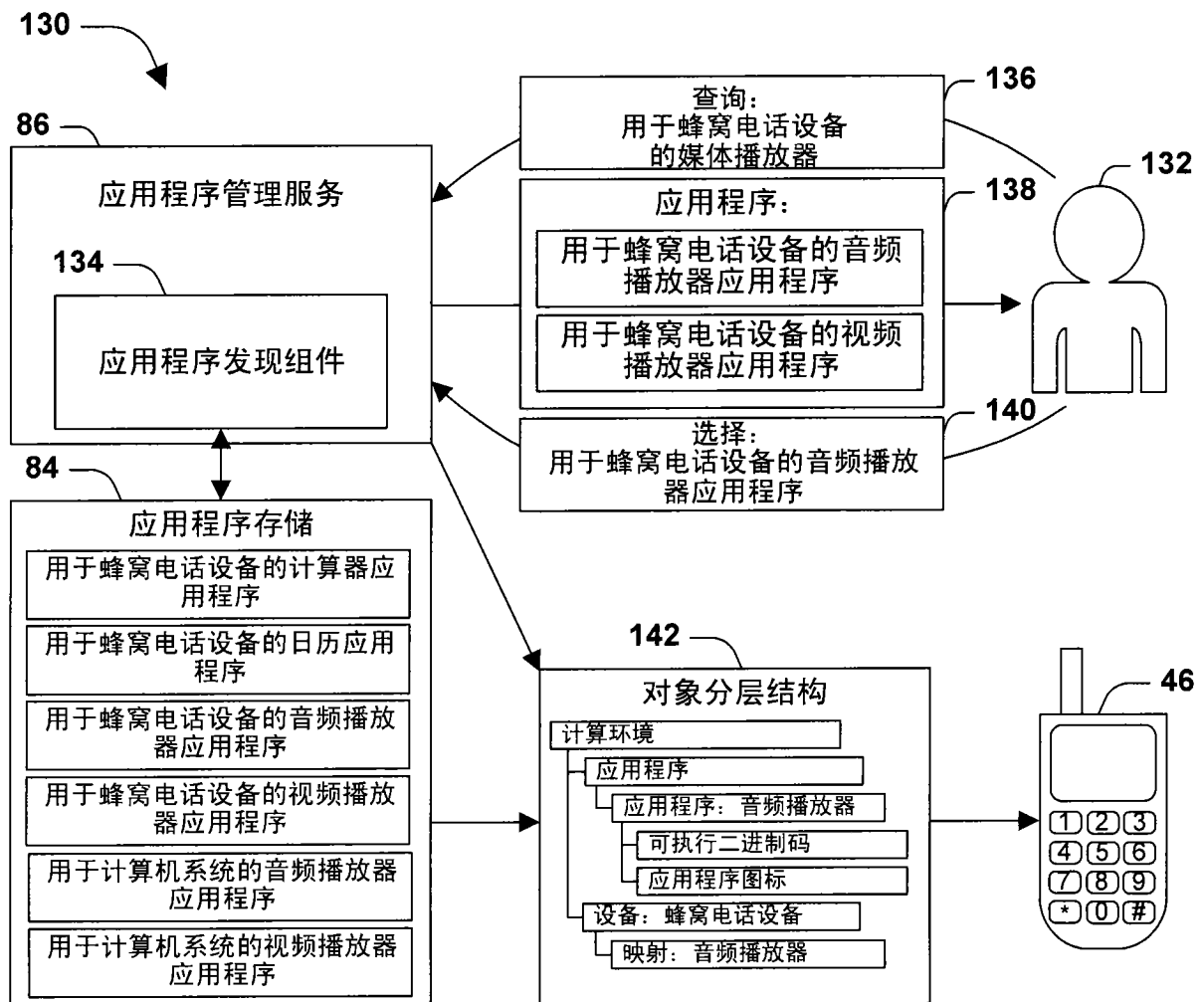


图 8

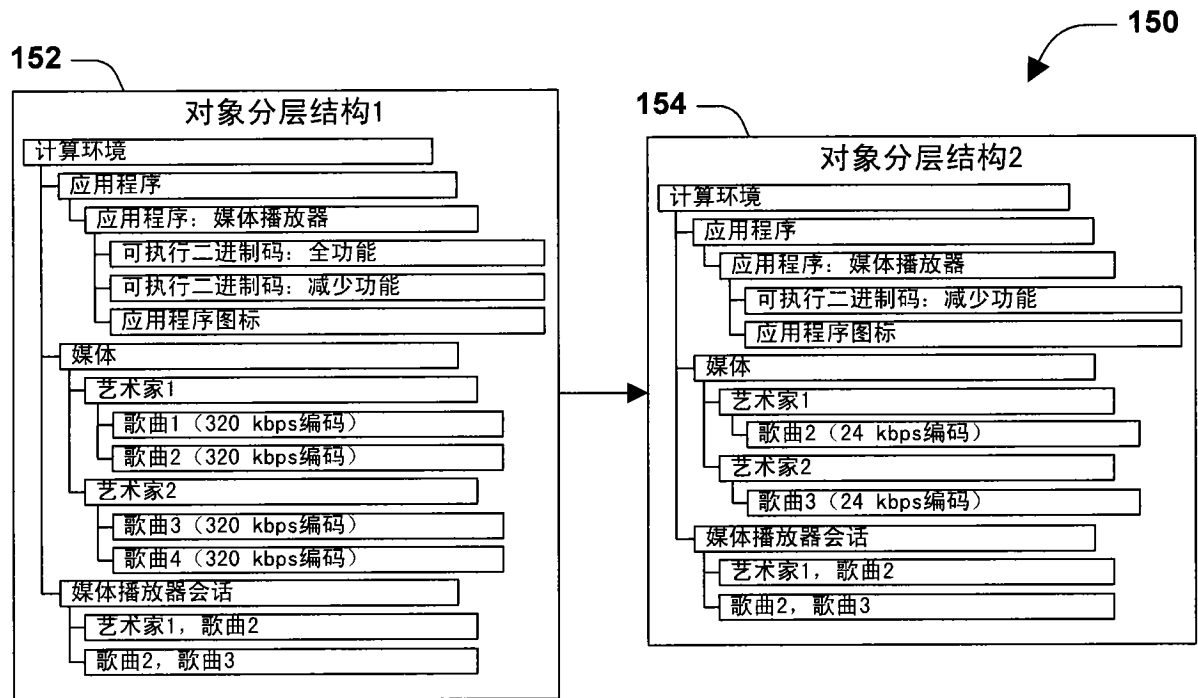


图 9

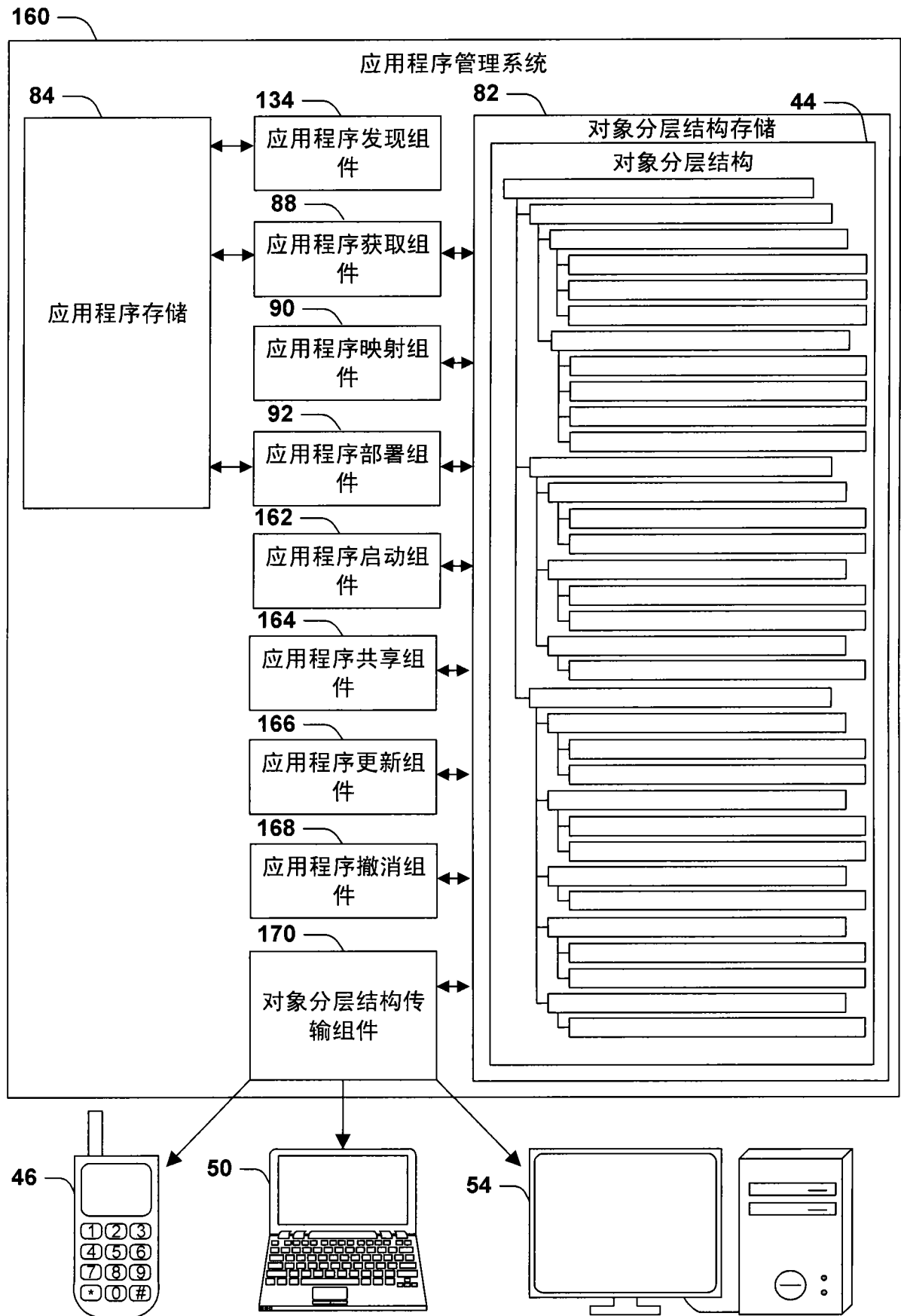


图 10

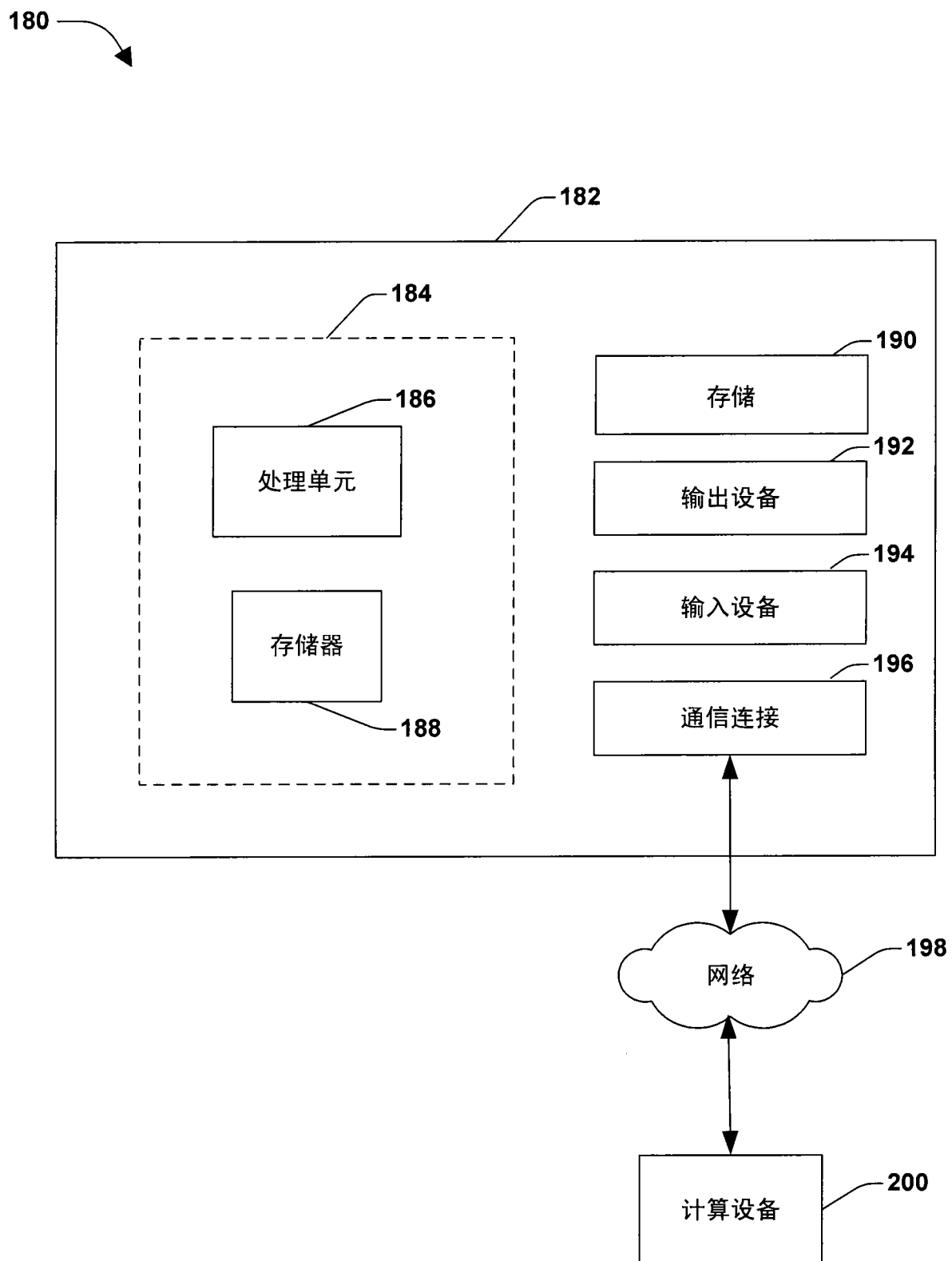


图 11