



(12) 发明专利

(10) 授权公告号 CN 101040552 B

(45) 授权公告日 2010.12.01

(21) 申请号 200680000981.3

(22) 申请日 2006.03.31

(30) 优先权数据

60/672, 096 2005. 04. 18 US

(85) PCT申请进入国家阶段日

2007. 04. 05

(86) PCT 申请的申请数据

PCT/CA2006/000481 2006.03.31

(87) PCT申请的公布数据

W02006/110991 EN 2006. 10. 26

(73) 专利权人 捷讯研究有限公司

地址 加拿大安大略省

(72)发明人 布伦杜沙·L·弗里奇

菲拉·比布尔

弗拉基米尔·布拉戈耶维奇

布赖恩·R·戈林

迈克尔·申菲尔德

卡门·B·维塔诺夫

(74) 专利代理机构 中科专利商标代理有限责任

公司 11021

代理人 王玮

(51) Int. Cl.

H04L 29/08 (2006.01)

G06F 9/445 (2006.01)

H04W 8/22 (2009.01)

H04W 88/02 (2009.01)

(56) 对比文件

CN 1535419 A, 2004. 10. 06, 说明书全文.

US 2004203693 A1, 2004. 10. 14, 说明书全文.

EP 1489501 A1, 2004. 12. 22, 说明书全文.

WO 9945451 A2, 1999.09.10, 说明书全文.

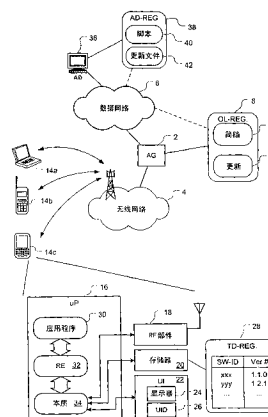
审查员 张剑

(54) 发明名称

控制软件版本更新的方法和系统

(57) 摘要

提供了用于控制软件更新的异步分发和安装的方法和系统,所述软件更新影响在无线网络的终端设备中安装的应用程序。由应用程序开发环境实施的版本号分配计划使终端设备的运行时间环境能够估计软件更新,以便识别潜在的兼容性问题并控制该更新的安装。



1. 一种控制无线网络的多个终端设备上软件更新的异步安装的方法,所述方法包括由主持对应终端设备的应用程序网关执行的步骤:

响应所述软件更新来接收更新通知消息,所述更新通知消息包括:

唯一地识别受所述更新影响的应用程序的软件标识符;

与所述软件更新相关的版本号;以及

所述对应终端设备可访问的数据网络中的更新脚本的地址,所述更新脚本适于在所述对应终端设备上安装所述软件更新;

在更新注册处记录所述更新通知消息;以及

向受到所述软件更新影响的终端设备转发通知消息,所述通知消息包括软件标识符、版本号和更新脚本的地址。

2. 如权利要求1所述的方法,其中,所述数据网络是因特网协议(IP)网络,以及更新脚本的地址包含所述IP网络上的所述更新脚本的统一资源定位器(URL)地址。

3. 如权利要求1所述的方法,其中,在注册处记录所述更新通知消息的步骤包括步骤:在更新注册处至少存储软件标识符、版本号和更新脚本的地址。

4. 如权利要求1所述的方法,其中,转发所述通知消息的步骤包括步骤:

搜索包含识别安装在所述多个终端设备中每一个上的每一个应用程序的信息的简档注册处,以便识别受到所述软件更新影响的每一个终端设备;以及

向每一个识别的终端设备发送所述通知消息。

5. 如权利要求4所述的方法,其中,所述简档注册处包括多个简档,每一个简档与各自的终端设备相关,并且对于安装在所述对应终端设备中每个应用程序,所述简档还包括:应用程序的各自软件标识符以及表示所述安装的应用程序的版本的当前版本号,以及其中搜索简档注册处的步骤包括步骤:识别更新的软件标识符与终端设备中安装的应用程序的标识符相匹配的每一个终端设备。

6. 如权利要求5所述的方法,还包括步骤:识别与所述软件更新相关的版本号的至少一个字段大于所述当前版本号的相应字段的每一个终端设备。

7. 如权利要求1所述的方法,其中,转发通知消息的步骤包括步骤:

从所述终端设备接收用于检查更新的消息;

访问与所述设备相关的简档,所述简档包含用于识别安装在所述终端设备上的软件的信息;

利用所述简档来搜索所述更新注册处,以便识别影响安装在所述终端设备上的软件的任意软件更新;以及

向所述终端设备发送关于每一个被识别的软件更新的通知消息。

8. 如权利要求7所述的方法,其中,对于安装在所述终端设备中的每一个应用程序,所述简档包括:各自的软件标识符和表示所述软件应用程序的版本的当前版本号,并且其中搜索更新注册处的步骤还包括步骤:

将与所述软件更新相关的版本号与所述当前版本号相比较;以及

识别所述相关版本号的至少一个字段大于所述当前版本号的相应字段的每一个软件更新。

9. 如权利要求1所述的方法,其中,所述脚本还适于针对不兼容的更新执行数据转换。

10. 一种控制无线网络的终端设备上的软件更新的异步安装的方法,所述方法包括步骤:

接收包含关于可用软件更新的信息的更新通知消息,所述更新通知消息包含:

唯一地识别所述更新影响的应用程序的软件标识符;

与所述软件更新相关的版本号;以及

所述终端设备可访问的数据网络中的更新脚本的地址,所述更新脚本适于在所述终端设备上安装所述软件更新;

利用所述更新版本号确定所述软件更新的兼容性;

利用所述地址访问所述更新脚本;以及

执行所述更新脚本。

11. 如权利要求 10 所述的方法,其中,确定所述软件更新的兼容性的步骤包括步骤:将所述更新版本号与受影响的应用程序的当前版本号相比较。

12. 如权利要求 11 所述的方法,其中,按照包含与应用程序的多个方面的每一个分别对应的字段的版本号分配计划来定义更新和当前版本号,其中将更新版本号与当前版本号相比较的步骤包括步骤:从更新版本号的相应字段的数值中减去当前版本号的每一个字段的数值。

13. 如权利要求 12 所述的方法,还包括步骤:如果所述减法结果中任意字段的数值大于 1,则确定在软件更新的安装之前必须安装的至少一个其他软件更新。

14. 如权利要求 12 所述的方法,其中,所述版本号分配计划包含与所述应用程序的数据组成相对应的至少一个“数据”字段,并且如果减法结果中的“数据”字段的数值大于“0”,则确定软件更新的兼容性的步骤包括:确定所述软件更新可能需要转换存储在所述终端设备中的用户数据。

15. 如权利要求 12 所述的方法,其中,所述版本号分配计划包含与所述应用程序的消息组成相对应的至少一个“消息”字段,并且如果减法结果中的“消息”字段的数值大于“0”,则确定软件更新的兼容性的步骤包括步骤:确定所述软件更新可能需要转换存储在所述终端设备中的用户数据。

16. 如权利要求 12 所述的方法,其中,所述版本号分配计划包含与应用程序的运行时间环境相对应的至少一个“RE”字段,并且如果减法结果的“RE”字段的数值大于“0”,则确定软件更新的兼容性的步骤包括步骤:确定应用程序的运行时间环境必须在所述软件更新前进行更新。

17. 如权利要求 10 所述的方法,其中,所述脚本还适于针对非兼容更新执行数据转换。

控制软件版本更新的方法和系统

技术领域

[0001] 本发明涉及无线通信设备,尤其涉及一种控制无线终端设备的软件版本更新的方法和系统。

背景技术

[0002] 诸如移动电话、可无线通信的膝上型电脑和具有无线通信能力的 PDA、自助电话亭和双向寻呼机之类的无线终端设备,数量与种类都在快速地增长。这些设备上运行的软件应用程序也增加了其效用。例如,移动电话可以包含应用程序,用来检索城市范围内的天气情况,或者 PDA 可以包含应用程序来使用户能够购买杂货。这些软件应用程序利用了到网络的连通性以便向用户提供及时和有效的服务。

[0003] 正如本领域所公知的,软件应用程序开发者频繁地提出他们软件的新的和 / 或更新的版本。这些软件更新的发布可能非常频繁,例如,作为补丁发布出来以解决前面发布的软件的缺陷。例如,可以一年或两年发布一次重要升级,并且经常提供新的功能以便加强特定设备的效用。

[0004] 但是,尽管软件开发者可能很容易开发并且发布软件更新,而实际在所有受影响设备上执行更新却是非常复杂的。例如,在无线网络中,连通性经常是断断续续的,以至于当发布更新时特定的设备可能没有连接到网络。在这种情况下,就需要一些方法来使更新在稍后的时间能够被下载并安装。甚至当完成这些时,一些设备还可能缺乏资源(例如足够的内存)来下载和成功地安装特定的更新。在其他情况下,应用程序更新可能需要在应用程序被安装前就更新设备的控制器软件。在其它情况下,可能必须以特定顺序来下载和安装一系列应用程序升级。因此,例如,必须在用于纠正新特征中包括缺陷的诸多问题的服务补丁之前,安装提供新特征的应用程序升级。

[0005] 因此,始终非常需要用于控制向无线终端设备安装软件更新的方法和系统。

[0006] 发明内容

[0007] 据此,本发明的目的在于提供用于控制向无线终端设备安装软件更新的方法和系统。

[0008] 因此,本发明的一个方面提供了一种控制无线网络的多个终端设备上软件更新的异步安装方法。按照本发明,由主持(hosting)对应终端设备的应用程序网关响应软件更新来接收更新通知消息。更新通知消息包括:唯一地识别更新影响的应用程序的软件标识符;与软件更新关联的版本号;以及对应终端设备可访问的数据网络中更新脚本的地址。更新脚本适于在对应终端设备上安装软件更新。在更新注册处记录更新通知消息,并且通知消息被转发给受到软件更新影响的终端设备。通知消息包括:软件标识符、版本号和更新脚本的地址,以便终端设备可以访问并执行更新脚本从而安装软件更新。

[0009] 本发明的另一个方面提供了一种控制无线网络的终端设备上软件更新的异步安装方法。按照本发明,由终端设备的运行时间环境接收包括与可用软件更新有关的信息的更新通知消息。更新通知消息包括:唯一地识别更新影响的应用程序的软件标识符;与该

软件更新关联的版本号；以及终端设备可访问的数据网络中更新脚本的地址，更新脚本适于在终端设备中安装软件更新。使用更新版本号来确定软件更新的兼容性。其后，使用地址来访问更新脚本，并执行它来安装软件更新。

[0010] 本发明的另一个方面提供了一种方法，能够控制影响安装在无线网络的多个终端设备上的应用程序的软件更新的分发。按照本发明，版本号分配计划被定义为包括分别与应用程序多方面中每一方面相对应的字段。当发布初始应用程序加载时，定义了每一个字段的初始值。对于影响应用程序的每一个连续软件更新，识别软件更新影响的应用程序的每一个方面，并且递增各个字段的值。

[0011] 附图说明

[0012] 结合附图，根据以下详细的描述本发明更多的特点和优点将变得显而易见，附图中：

[0013] 图 1 是示意性地示出了网络系统的框图；

[0014] 图 2 是示意性地示出了按照本发明一方面的应用程序开发环境的结构和操作的框图；

[0015] 图 3 是示意性地示出了按照本发明一方面的公布软件升级的处理的消息流程图；

[0016] 图 4 是示意性地示出了按照本发明实施例在终端设备上安装软件升级的处理的消息流程图；以及

[0017] 图 5 是示意性地示出了按照本发明另一实施例在终端设备上安装软件升级的处理的消息流程图。

[0018] 应当注意，在所有附图中，相同的部分利用相同的参考数字标识。

具体实施方式

[0019] 本发明提供了用于控制无线终端设备上软件更新的分发和安装的方法和系统。以下仅作为示例，参考图 1-5 来描述本发明的实施例。

[0020] 参考图 1，按照本发明典型实施例的系统通常包含连接在无线网络 4 和数据网络 6（例如，因特网）之间的应用程序网关（AG）2。系统还具有在线注册处 8，包括：简档注册处 10，包含针对每一个订户的终端设备 14a、14b、14c 的信息表，用于识别分别存储在各个终端设备上的软件应用程序；以及更新注册处 12，包含识别任意可用应用程序更新的信息。

[0021] 按照申请人同时待审的美国专利公开 No. 2004/0215700 和 2004/0220998（这二者一并在此作为参考）所述的方式，AG2 通常进行操作，以便在与无线网络 4 相连的终端设备 14a、14b、14c 和通过数据网络 6 可访问的数据业务之间传达消息流。

[0022] 在线注册处 8 可以与 AG 2 共同驻留（co-resident）或者可以远离 AG 并由 AG 通过数据网络 6 进行访问。在图 1 的实施例中，在线注册处 8 包括简档注册处 10 和更新注册处 12。简档注册处 10 包含针对多个终端设备的每一个的简档。每一个简档都至少包含唯一地识别运行时间环境（RE）和安装在各自终端设备上的每一个应用程序的软件标识符（SW-ID）的列表。安装在终端设备上的每一个应用程序的各自“当前”版本号还可以与各自的 SW-ID 相关联地存储在在线注册处 8，或者可以存储在终端设备中。针对每一个软件更新，分离的脚本注册处 40 包含设计用于执行终端设备上的软件更新的一个或多个脚本。

[0023] 通常，终端设备可以是多种软件控制的无线设备中的任意一种，包括但不限于：无

线启用的膝上型计算机 14a、移动或蜂窝电话 14b、具有无线通信功能的 PDA 14c、自助电话亭以及双向寻呼机。从图 1 中可以看出,这些设备通常包括:与用于无线通信的 RF 部件 18 相连的微处理器 16、存储器 20(至少其一部分通常是非易失性的)、以及包括显示器 24 和一个或多个用户输入设备 (UID) 26(例如键盘、指轮、输入笔、麦克风等)的用户接口 (UI) 22。微处理器 16 在软件控制下进行操作,以提供终端设备的功能。优选地,以一种分层模型来设计软件,其中 RE 32 在应用程序软件 30 和终端设备的本机语言 34 之间进行翻译来控制终端设备的硬件,并与数据业务进行通信。这种分层软件模型及其操作的方式可以从申请人同时待审的美国专利公开 No. 2004/0215700 和 2004/0220998 中得到。RE 还可以保持终端设备注册处 28(图 1 中表示为“TD-REG”),用来识别终端设备上安装的每一个应用程序以及每一个应用程序的当前版本号。下面将详细描述能够向终端设备异步分发和安装软件升级的 RE 的操作。

[0024] 如申请人同时待审的美国专利公开 No. 2004/0215700 和 2004/0220998 中所述,AG 2 的操作能够使软件应用程序在终端设备中执行以便与数据网络 6 提供的业务(未示出)进行通信。例如,这个操作可以包括访问并下载来自与数据网络 6 相连的后端数据源(未示出)的文件。如图 1 中可以看出,以及下面更详细所述,应用程序开发者 (AD) 36 还可以通过数据网络 6 分发和支持新的或更新的软件。例如,可下载的应用程序软件和安装脚本可以存储在能被用户(直接或间接地)通过数据网络 6 访问的应用程序开发者注册处 38。

[0025] 应用程序开发环境

[0026] 现在参考图 2,应用程序开发者按照本领域通常公知的方式,使用运行在计算设备上的应用程序开发环境 (ADE) 50 的应用程序开发工具箱 (ADT) 52 来对应用程序软件进行编码、测试和调试。计算设备可以是连接到或者可连接到数据网络或其它网络工作站的个人计算机或膝上型计算机。再次按照本领域公知的方式,该相同的 ADE 还可以用于开发应用程序的后续更新。按照本发明,ADE 50 还包括版本号分配 (versioning) 模块 70,基于编码、测试和调试的处理中应用程序源代码发生的改变来自动地分配版本号。版本号分配模块 70 还能够用于产生更新脚本 60,当在终端设备中执行所述更新脚本 60 时,更新脚本 60 将下载和安装终端设备上的更新。

[0027] 例如,版本号分配模块 70 能够用于识别以下任何之一:

[0028] • 现有数据组成(诸如数据结构)的改变,即,通过添加或删除字段,或改变字段类型的定义;在全局变量定义和枚举的改变;

[0029] • 现有消息的改变,即,通过添加或删除字段,或改变字段类型定义;

[0030] • 现有应用程序逻辑的改变;

[0031] • 要被添加到应用程序的新数据组成、消息或应用程序逻辑。

[0032] 在每一种情况下,版本号分配模块 70 检测到的改变和添加均与应用程序的“当前”版本相关(也就是说,具有所安装的任意后续发布的更新的初始版本)。根据需要,可以通过利用修订日志 54 在编辑处理期间实时跟踪应用程序开发者的动作(例如键击),或者通过比较应用程序源代码的“过去”和“之后”版本,或者通过将应用程序的更新版本与该应用程序的之前版本进行比较来确定发生了什么改变的任意其它方式,来检测改变。在任何情况下,版本号分配模块 70 识别随后被写入一个或多个更新文件 58 的新的和/或修订

的数据组成、消息、以及应用程序逻辑。然后,这些更新文件 58 可以被保存到更新文件注册处 42,优选地,如图 1 和 2 所示,更新文件注册处 42 驻留在应用程序开发者注册处 38。可选地,在如图 2 所示的实施例中,修订日志 54 跟踪由 AD 工具箱 52 对应用程序源代码做的改变。可以理解,版本号分配模块 70 可以包含修订日志 54,或者修订日志 54 可以是 ADE50 中的独立模块。

[0033] 另外,如下面更详细地描述,可以生成更新脚本 60 来控制终端设备下载和安装更新文件 58。更新脚本 60 可以被保存到脚本注册处 40,优选地,如图 2 所示,脚本注册处 40 驻留在应用程序开发者注册处 38。

[0034] 如果需要,可以提供脚本编辑模块或更新脚本编辑器 56,以使应用程序开发者可以手动编写更新脚本 60,也可以回顾和编辑自动生成的更新脚本。

[0035] 可以便利地以结构化语言(诸如 Java, JavaScript 或 XML)来开发更新脚本 60,以使设备可以在更新处理期间,通过数据网络 6 来访问一个或多个后端数据源。这使得更新脚本 60 能够在执行期间根据需要来访问和下载更新文件 58。由于更新脚本 60 能够在执行期间将需要的更新文件 58 从后端数据源“拉出(pull)”,这促进了更新的异步分发。此外,在特定的实现方式中,用于升级的通知消息可以包含足够的信息使终端设备中的 RE 能够自己拉出应用程序更新。

[0036] 版本号分配计划

[0037] 如上所述,版本号分配模块 70 自动地向初始版本和应用程序的每一个更新分配各自的版本号。在初始版本的情况下,可以使用任意需要的“初始”版本号。但是,在初始版本之后,基于该更新所做的改变的类型,向每个后续的更新分配自动生成的版本号。这个设置具有很多优点。例如,由于由版本号分配模块 70 分配每一个版本号,因此版本号之间的连续性是强制的。例如,这意味着终端设备的 RE 可以使用更新的版本号来确定是否必须首先安装任意其它的更新。另一个优点在于,终端设备的 RE 可以在试图安装更新之前,估计更新的版本号以检测潜在的兼容性问题。

[0038] 通过将版本号格式化为多个字段的版本号分配计划来启用这些功能,其中每一个字段分别表示应用程序的不同方面。例如,版本号分配模块被设计成检测现有数据组成、消息或逻辑的改变,以及新数据组成、消息或逻辑的添加。理论上说,这些因素的每一个都可以用版本号的各个字段表示。但是,实际上却发现使用“数据·消息·特征”组合排列的三字段计划可以得到令人满意的性能,如下面的表 1 所示。

[0039] 表 1

字段	描述
数据	现有数据定义的改变。例如,这可以包括:数据组成的改变(例如添加或去除数据字段、改变字段类型的定义);永久的全局变量;或枚举。
消息	应用程序使用的现有消息的改变。例如,这可以包括:消息组成的改变(例如添加或去除字段、改变字段类型定义、或改变映射)。
特征	应用程序特征的添加。例如,这可以包括:添加或改变应用程序逻辑、筛选(screen)和/或全局;新消息的添加;和/或数据组成或字段的添加。

[0040] 利用这个版本号分配计划,可以针对应用程序的初始版本向版本号的每个字段分配初始值(例如 D = 1, M = 1, F = 0)。之后,对于每一个更新版本,版本号分配模块通过递增应用程序字段的数值,自动地产生分别用于更新的各个版本号。

[0041] 例如,如前所述,考虑携带版本号“1.1.0”被发布的应用程序。在初始发布之后,应

用程序开发者提出应用程序更新,修改了现有的数据字段并添加了新的应用程序逻辑。通过递增数据和特征字段,这些改变将反映在更新的版本号中。因此,第一次更新的版本号将是 2.1.1。在首次更新的发布之后,它的版本号 (2.1.1) 变为应用程序的“当前”版本号,并与下次发布的应用程序更新进行比较。因此,例如考虑第二个应用程序更新,修改了现有消息的格式。通过递增消息字段,在第二个更新的版本号中反映这个改变,因此第二个更新版本号将是 2.2.1。

[0042] 应当理解,这个模式可以持续任意次数的更新,每一次都可能影响 应用程序的任意或多个方面。还将理解,可以扩展版本号分配计划以提供更精细的粒度。例如,可以利用分别表示新数据、消息或应用程序逻辑的添加的字段集合来替代特征字段。在另一个示例,可以将一个字段添加到版本号上,以表示是否必须在应用程序的安装前更新 RE。

[0043] 还可以理解,本发明并不限于应用程序本身。例如,版本号分配计划以及此处描述的更新方法可以等同地应用于 RE 本身,由此可以控制 RE 的更新。

[0044] 异步软件分发

[0045] 参考图 3,当应用程序开发者 (AD) 36 发行 (初始软件加载或更新的) 软件版本时,版本号分配模块分配版本号,并将更新脚本和更新文件存储于 AD 注册处 38。然后,AD 36 向 AG 2 发出更新通知消息。优选地,更新通知消息包括唯一地识别应用程序的软件标识符 (SW-ID)、版本号、以及到存储在脚本注册处 40 中的更新脚本的链路 (例如 URL)。当 AG 2 从 AD 36 接收到更新通知消息时,AG 2 通过在更新注册处 12 存储软件 ID、版本号和脚本 URL 来记录更新。一旦更新被 AG 2 记录下来,就可以按照多种方式实现向用户的终端设备异步分发。下面参考图 4 和 5 描述两种有代表性的分发方案。

[0046] 图 4 示出了一种异步分发方案,例如,由 AG 2 响应从 AD 36 接收到的更新通知消息来发起所述方案。因此,AG 2 使用软件 ID (例如包含在更新通知里) 来搜索简档注册处 10。搜索返回识别了其已经安装了应用程序的所有终端设备的信息 (例如设备 ID)。然后,AG 2 可以生成通知消息并将通知消息发送到所识别的每一个终端设备。例如,通知消息可以包含更新的软件 ID 和版本号、以及到存储于应用程序开发者注册处 38 的脚本部分 40 中的更新脚本的链路 (例如,URL)。

[0047] 当接收到通知消息时,运行时间环境 (RE) 可以从消息中提取软件 ID 和版本号,并且使用这个信息来确定更新是否可以被安全地安装在终端设备上。这个估计可以采用以下兼容性检查的形式:

[0048] • 将“新”版本号与保存在终端设备注册处 (TD-REG) 28 中的当前 版本号进行比较,以识别在更新中将改变应用程序的哪些方面。例如,可以通过将新的版本号和当前版本号逐字段相减的方式来实现这项功能。例如,考虑当前版本号是“2.2.1”以及包含在通知消息中的新版本号是“2.3.2”的情况。从新版本号中逐字段地减去当前版本号得到“0.1.1”,这表示更新涉及:现有数据组成不改变;至少一个现有消息发生改变;至少增加了一个新的特征。

[0049] • 确定是否必须在通知消息标识的“当前”更新之前安装任意插入的更新。这可以通过上述的减法结果来完成。具体地,如果减法结果的任意字段都有大于“1”的数值,那么必须在“当前”更新之前安装至少一个更新。

[0050] 一般而言,新特征的添加不会产生任何兼容性问题。但是,现有数据组成或消息的

改变可能会有兼容性问题,这是因为存储在存储器中的用户数据有可能与修订的数据和/或消息定义不兼容。因此,更新的安装将需要所保存的数据的转换,并且执行这种转换的能力可能会受终端设备的硬件性能的限制。此外,数据转换会存在一些数据可能被破坏或丢失的风险,这样用户就可能宁愿不安装更新,即使所需要的转换功能是终端设备性能所具有的。

[0051] 因此,只要 RE 确定更新影响了现有数据组成和/或现有消息,则 RE 就可以向用户提供警告消息,表示更新是可用的,但是其安装可能会引起数据丢失或破坏。然后,该用户可以选择是否安装更新。可选地,由应用程序开发者使用 AD 工具定义的更新脚本可以对于不兼容的更新执行数据转换。

[0052] 当(或者如果)该用户选择安装更新时,或者如果 RE 确定没有兼容性问题(即,更新仅添加新特征),则 RE 可以通过打开包含在更新通知消息中的链路(URL)来发起更新的安装,并由此访问和下载更新脚本。当从应用程序开发者注册处(AD-REG) 38 中的脚本部件 40 成功地下载更新脚本时,RE 可以装入脚本,然后所述脚本控制从应用程序开发者注册处 38 的更新文件部分 42 下载和安装更新文件。

[0053] 当成功地安装了更新时,则 RE 使用在更新通知消息中接收的更新版本号,更新存储于终端设备注册处中的应用程序的“当前”版本号,并向 AG2 发出更新完成消息。当接收到更新完成消息时,AG 2 利用新版本号更新设备简档,由此表示软件更新已经成功地安装在终端设备 14b 上。

[0054] 图 4 所述方案的局限在于 AG 2 通过向其上安装了受影响的软件的每一个终端设备发送通知消息来发起更新分发方案(例如,响应从 AD36 接收到更新通知消息)。这可能导致不愿看到的大量通知消息涌入网络,其可能加重无线网络带宽的负荷。此外,当 AG 发送通知时,一些终端设备可能没有被连接,结果“断开”的终端设备可能错过该更新。这些问题可以由下面参考图 5 描述的异步分发方案来解决。

[0055] 在图 5 示出的分发方案中,由终端设备触发异步分发。在示出的示例中,触发事件在于当终端设备登录到 AG 2 上时,尽管也可以使用其它事件。例如,RE 可以向 AG 2 发送消息以便按照预定的时间表来检查更新,或者当将应用程序装入终端设备 14b 时来检查更新。无论如何,AG 2 通过访问终端设备的简档来响应终端设备 14b,以便识别安装在终端设备 14b 上的每一个应用程序和当前的版本号。然后,这个信息用于搜索更新注册处 12,以识别影响终端设备 14b 的任意记录的更新以及相应的更新版本号。然后,当前和更新版本号之间的比较使 AG 能够确定终端设备 14b 是否还存在没有被安装的任意更新。

[0056] 然后,AG 2 针对每一个未安装的更新定制适当的更新通知消息,并将其发送给终端设备 14b。然后,终端设备 14b 检查更新通知消息和安装更新的后续处理符合参考图 4 描述的上述处理,也就是说,该终端设备 14b 检查兼容性,然后通过打开链路(URL)从 AD-REG 38 下载脚本来安装该更新。在该终端设备上执行该脚本使得能够访问和下载存储在 AD-REG 38 中的更新文件。当更新完成后,终端设备通过向 AG 通知软件 ID 和版本号来信号通知表示更新的完成。然后,该 AG 通过向简档注册处 10 通知软件 ID 和版本号来更新简档注册处 10 中的简档文件。

[0057] 上述本发明的实施例意欲仅作为示例。因此,仅由所附的权利要求来限制本发明的范围。

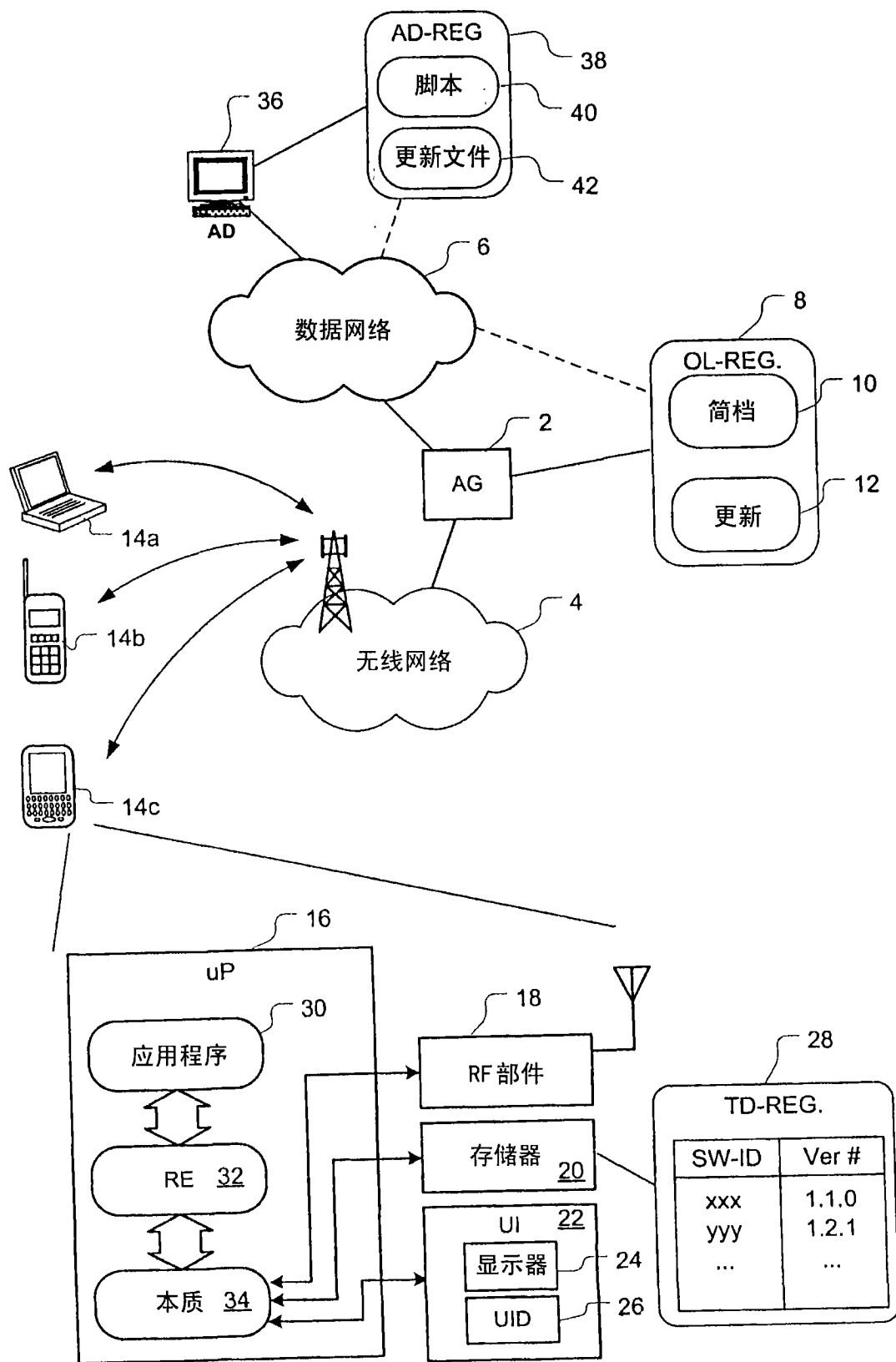


图 1

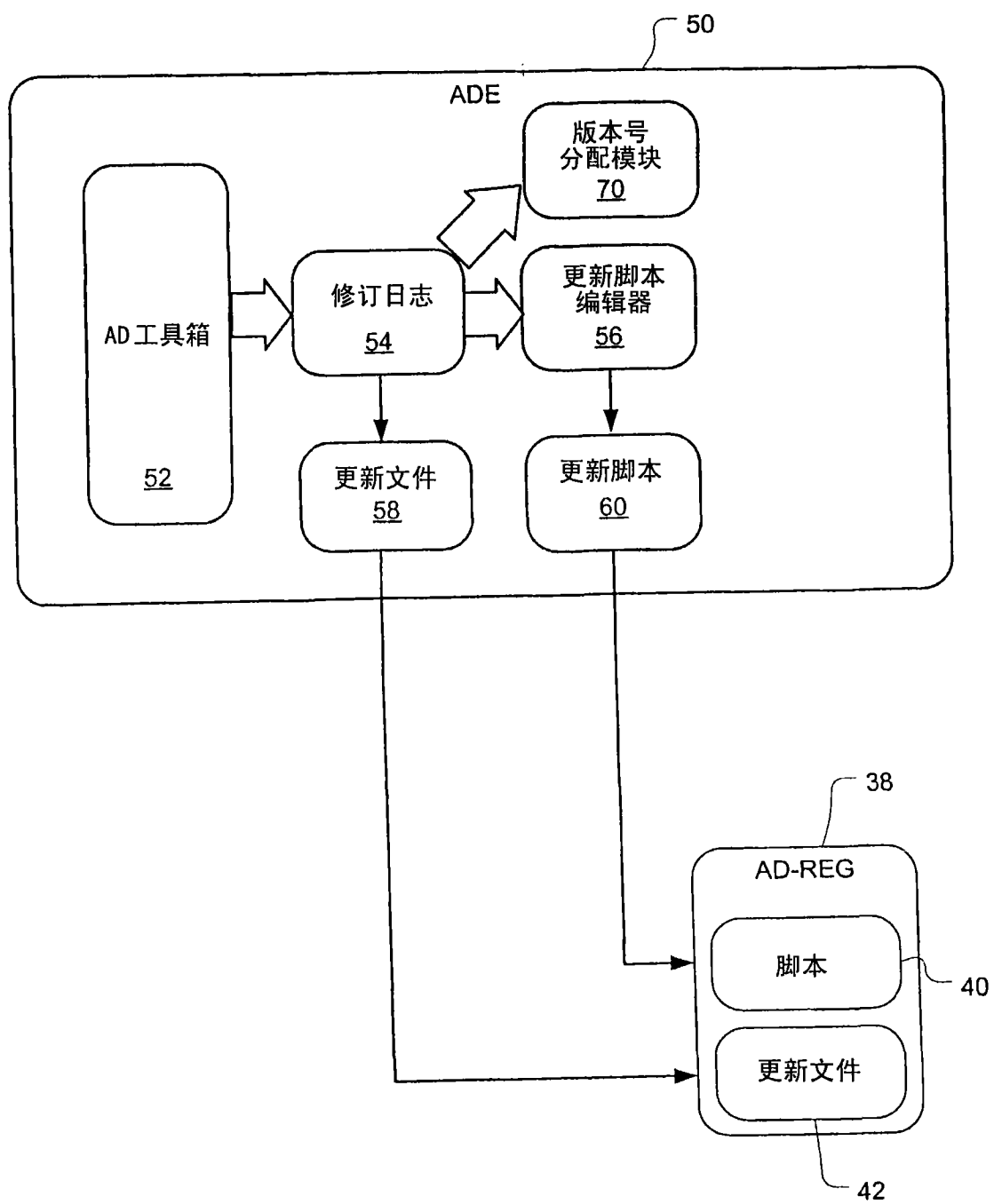


图 2

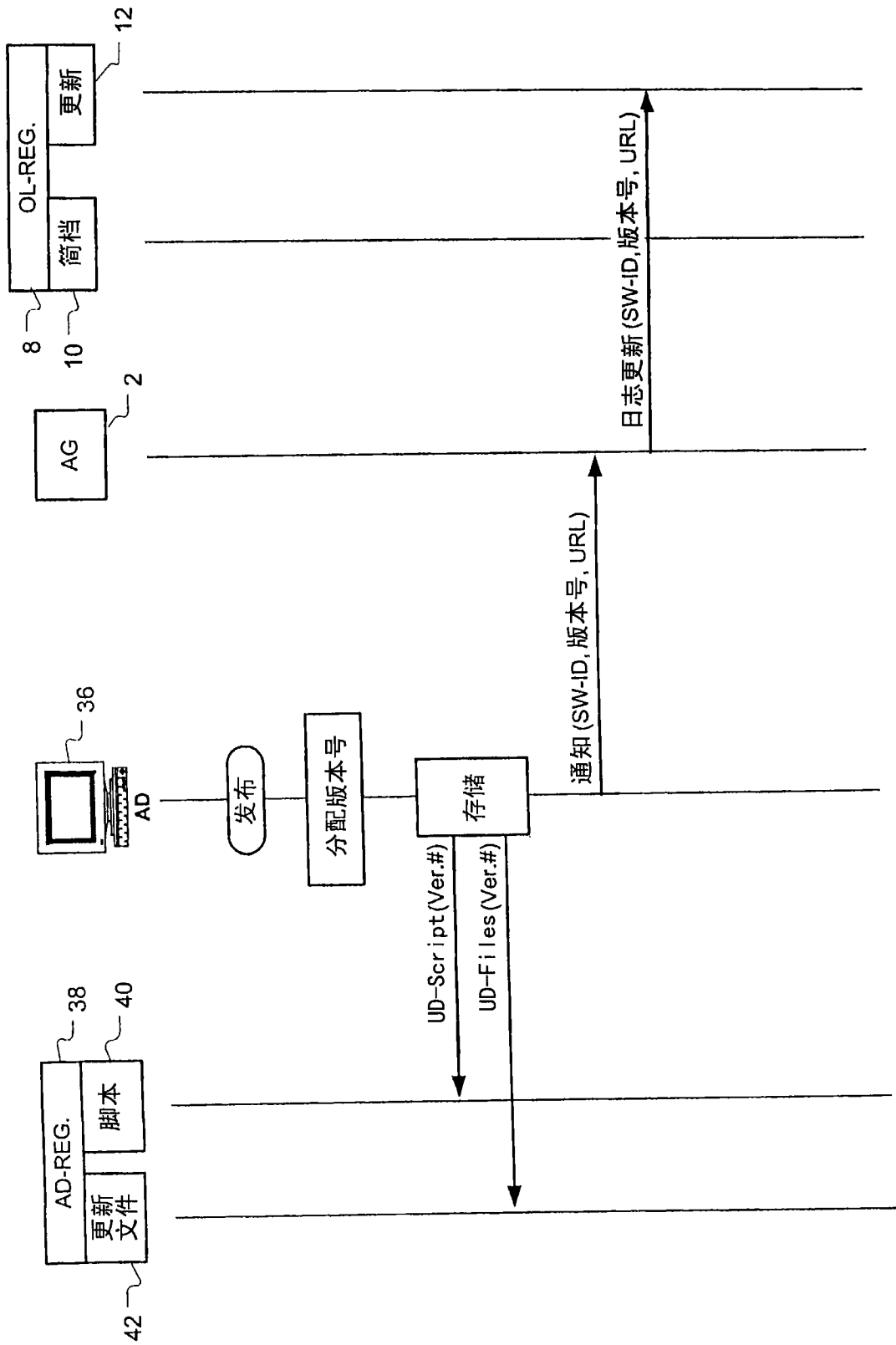


图 3

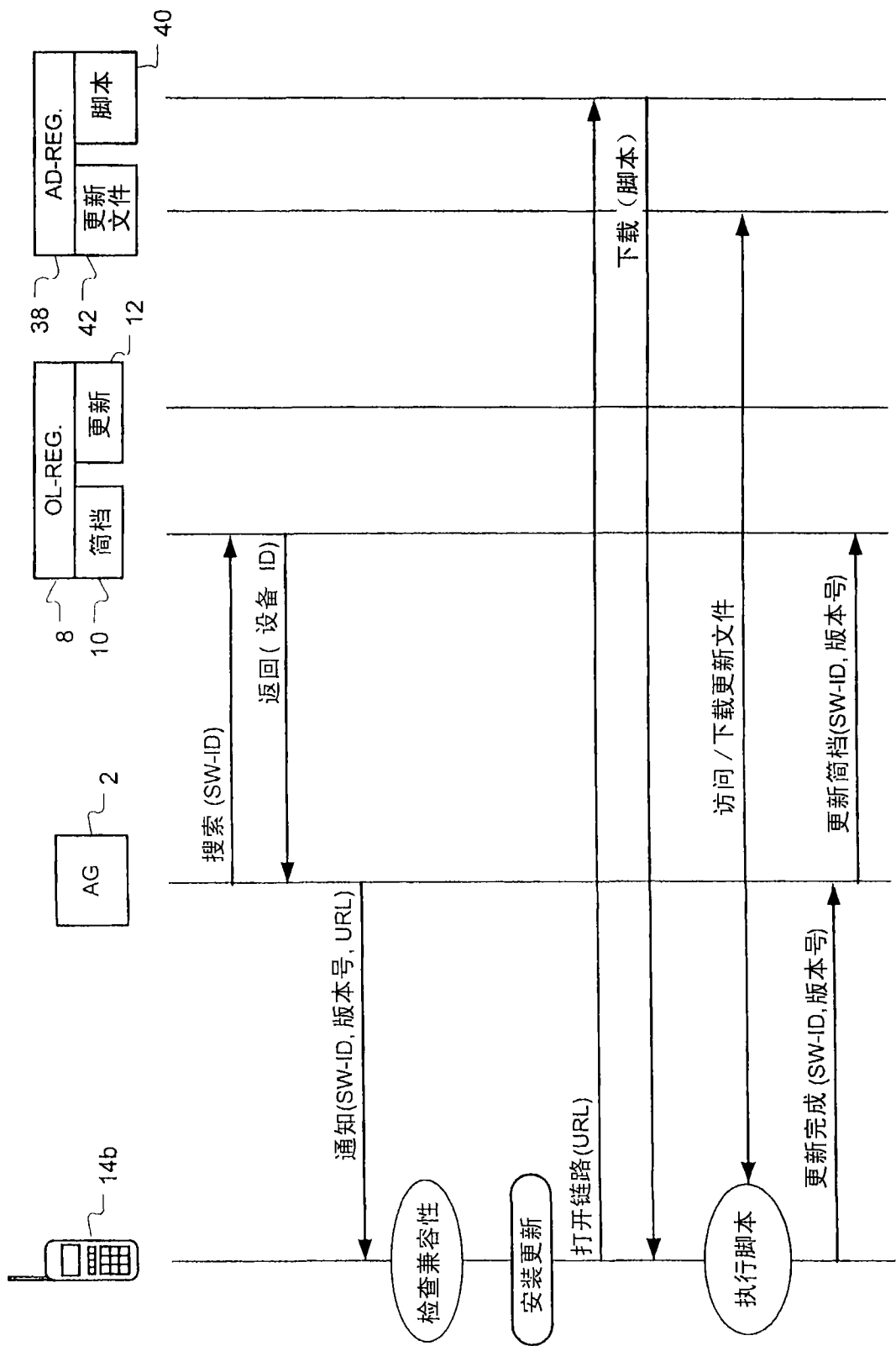


图 4

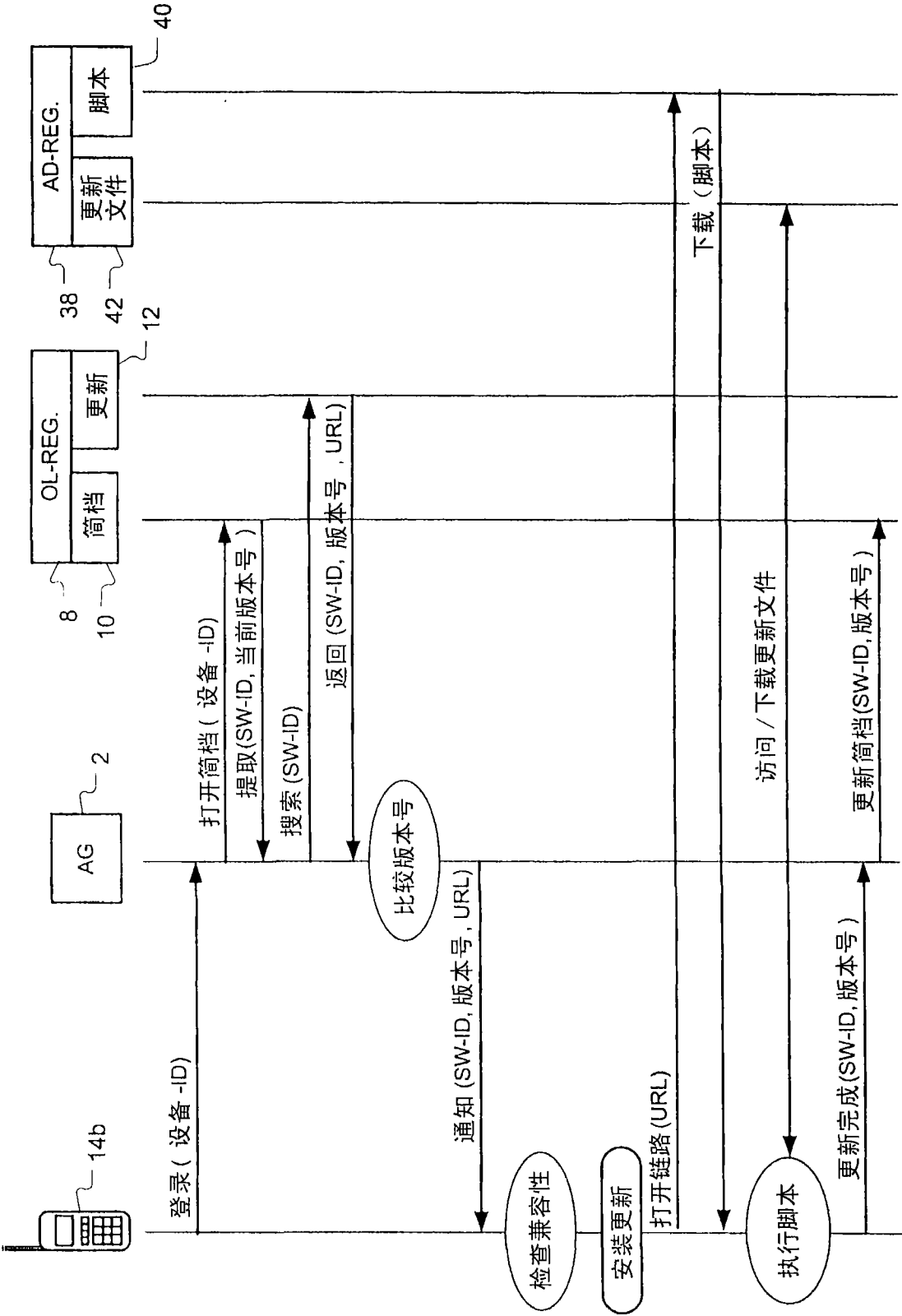


图 5