[19] 中华人民共和国国家知识产权局

[51] Int. Cl[7]
G06F 9/44

[12] 发明专利申请公开说明书

[21] 申请号 200410062863.8

[43] 公开日 2005 年 2 月 2 日

[11] 公开号 CN 1573692A

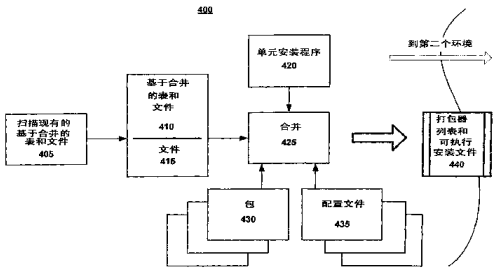
[22] 申请日 2004.6.21
[21] 申请号 200410062863.8
[30] 优先权
[32] 2003. 6.20 [33] US [31] 10/600,178
[71] 申请人 微软公司
地址 美国华盛顿州
[72] 发明人 J·K·哈塞尔登 A·沙马
[74] 专利代理机构 上海专利商标事务所
代理人 李家麟

权利要求书 3 页 说明书 9 页 附图 6 页

[54] 发明名称 基于依存关系的包部署

[57] 摘要

提供包和它们的依存物的部署。 在示意性的实施例中，接口用于暴露包的组件并确定这些组件的依存关系。 在操作中，对包的组件执行扫描来识别组件依赖的那些元素。 扫描的结果是在将被部署的包中找到的组件的依存关系的列表。 典型接口进一步用于聚合及合并用于部署的包的所有组件和它们的依存关系。 一旦经部署，接口在新部署的环境中执行，以展开包并在新环境中放置组件和它们的依存关系。 使用确定为包组装和部署的部分的依存关系，接口确保正确地配置并正确地操作组件。



ISSN1008-4274

1. 一种部署组件的方法，其特征在于，所述方法包括：

通过组件定义依存关系，所述组件向合作的接口暴露它们的依存关系；

产生所定义的依存关系的列表；

识别和所定义的依存关系关联的文件；及

处理所述依存关系列表和所识别的文件来产生可配置的包。

2. 如权利要求 1 所述的方法，其特征在于，进一步包括发送可部署的包到合作的环境。

3. 如权利要求 1 所述的方法，其特征在于，进一步包括合并所述依存关系列表、所识别的文件及缺省组件元素来产生所述可部署的包。

4. 如权利要求 1 所述的方法，其特征在于，进一步包括扫描所述组件来识别缺省元素和非缺省元素。

5. 如权利要求 1 所述的方法，其特征在于，进一步包括提供安装程序组件，用于包括在所述可部署的包中。

6. 如权利要求 5 所述的方法，其特征在于，进一步包括通过在计算环境中安装可部署的包来校验可部署的包。

7. 如权利要求 1 所述的方法，其特征在于，进一步包括提供配置文件，用于包括在所述可部署的包中。

8. 如权利要求 1 所述的方法，其特征在于，所述处理步骤进一步包括合并所述依存关系列表、所识别的文件和缺省组件元素来产生可部署的包。

9. 如权利要求 8 所述的方法，其特征在于，进一步包括提供合并模块，用于合并所述依存关系列表、所识别的文件和缺省组件元素来产生可部署的包。

10. 一种计算机可读媒体，其特征在于，所述媒体包括计算机可读指令，所述指令用于：

通过组件定义依存关系，所述组件向合作的接口暴露它们的依存关系；

产生所定义的依存关系的列表；

识别和所定义的依存关系关联的文件；及

处理所述依存关系列表和所识别的文件来产生可配置的包。

11. 一种用于创建要部署的包的数据结构，其特征在于，所述数据结构包括：

包含表示包组件的数据的数据字段；

包含通过扫描第一个字段来识别包组件所依赖的项目从包含表示包组件的数据的字段衍生得到的依存关系数据的数据字段；及

用于划定数据结构终止处的字段。

12. 如权利要求 11 所述的数据结构，其特征在于，进一步包括通过合并组件依存关系与表示包组件的数据从包含依存关系的数据字段衍生得到的字段。

13. 如权利要求 12 所述的数据结构，其特征在于，所述包含合并的数据的字段包括表示有关如何部署所述包的配置信息的数据。

14. 如权利要求 11 所述的数据结构，其特征在于，进一步包括控制安装程序安装所述包的字段。

15. 一种部署有依存关系的包的系统，其特征在于，所述系统包括：

表示包组件依存关系的依存关系数据；

用于合并依存关系数据与包组件来产生可部署的包的合并模块。

16. 如权利要求 15 所述的系统，其特征在于，进一步包括安装程序，所述安装程序由合并模块合并来产生所述可部署的包。

17. 如权利要求 16 所述的系统，其特征在于，进一步包括配置信息，所述配置信息由合并模块合并来产生所述可部署的包。

18. 如权利要求 17 所述的系统，其特征在于，进一步包括缺省包组件，所述缺省包组件由合并模块合并来产生所述可部署的包。

19. 如权利要求 18 所述的系统，其特征在于，进一步包括发送方法，用于发送所述可部署的包到合作的计算环境。

20. 如权利要求 19 所述的系统，其特征在于，进一步包括校验包的正确部署的校验方法。

21. 一种在有依存关系的组件部署中使用的系统，其特征在于，所述系统包括：

确定组件依存关系的方法，所述组件向所述方法暴露所述依存关系；及

用于合并组件依存关系与组件以产生可部署的包方法。

22. 如权利要求 21 所述的系统，其特征在于，进一步包括安装方法，所述安装方法与合并方法一起用于安装所述可部署的包。

基于依存关系的包部署

技术领域

本发明涉及计算应用组件的部署，更特别地，涉及使用依存关系的包部署，以使得同时部署包的相关组件从而确保正确的操作。

背景技术

随着计算和通讯技术的发展，出现了市场的全球化并保持这一势头，公司和政府企业的扩展同样如此。大跨度数据和信息共享的地理限制正成为历史。再者，随着这些技术发展，曾经安排用于独立机器的计算应用现在通常跨越大型计算环境网络和平台进行部署。随着共享更多的数据和计算应用，产生了管理和控制这些数据和应用的部署来确保完整性和可重用性的需要。

很多组织需要集中数据来改进公司决策。然而，他们的数据可能以各种格式存储并存储在不同的位置。数据转换服务（DTS）通过提供一组允许抽取、转换来自不同源的数据并将它们集成到由 DTS 连接性支持的单个或多个目的地中的工具来处理这个至关重要的业务需求。通过使用 DTS 工具以图形方式建立 DTS 包或通过用 DTS 对象模型对包编程，可以创建对特殊的业务需求定制的自定义数据迁移方案。

DTS 包是连接、DTS 任务、DTS 转换及用 DTS 工具或以编程方式对各种合作的计算应用或对结构化的存储文件组装的流程限制的有组织的集合。每个包通常包含在运行包时顺序或并行执行的一个或多个步骤。当执行时，包连接到正确的数据源、复制数据和数据库对象、转换数据，并通知其他用户或事件处理。可以对包进行编辑、密码保护、定时执行并根据版本来检索。

在企业当中，经常性地移动包是很常见的。当移动包时，在多数情况下，需要进行系统升级或从开发机器到测试机器的系统迁移。在过去，不进行很多次试错很难成功地实现这样的处理。特别地，DTS 是可嵌入组件结构且第三方可以创建组件，使得立即确定哪些组件属于哪个包很困难。再者，当前实现 DTS 配置仅安装缺省的组件，且使得确定并部署包所依赖的所有项目很困难。

目前，希望部署包的人只能手动地聚合给定包的所有组件的所有依存关系。由于部署包的人必须识别特殊包的所有组件然后识别任何特殊组件依赖的所有文件、表、其他任务，和/或其他包，这样的处理耗时费力。这样的信息对包的成功部署至关重要，因为如果存在缺失的元素（即，任何被忽视的依存关系），已部署的包将不能在新部署的环境中正常工作。一旦经收集和聚合，组装和部署包的人必须合并所有的组件（如，任务、文件、表和其他依存关系）到用于部署的包中。一旦经部署，必须在新部署的环境中反向地执行这一处理来确保该包能正常工作。简单地，现有的实践是低效的且留下了在其中可能产生并传播错误的很大空间。

根据上述，应理解，存在对超过现有技术的系统和方法的需要。

发明内容

提供包和它们的依存物的部署。在示意性的实施例中，接口用于暴露包的组件并确定这些组件的依存关系。在操作中，对包的组件执行扫描来识别组件依赖的那些元素。扫描的结果是在将被部署的包中找到的组件的依存关系的列表。典型接口进一步用于聚合及合并用于部署的包的所有组件和它们的依存关系。一旦经部署，接口在新部署的环境中执行，以展开包并在新环境中放置组件和它们的依存关系。 使用确定为包组装和部署的部分的依存关系，接口确保正确地配置并正确地操作组件。

在下面说明本发明的其他方面。

附图说明

引用附图进一步说明基于依存关系的包部署的系统和方法，其中：

图 1 和图 2 为适合于本发明的典型计算环境的示意图，其中图 2 展示典型联网计算环境；

图 3 为根据在此所述的系统和方法使用基于依存关系的包部署的典型计算环境的方框图；

图 4 为当执行基于依存关系的包部署时的典型数据流的方框图；

图 5 为执行用于准备用于部署的包的典型处理的流程图；及

图 6 为执行用于展开已部署的包的典型处理的流程图。

具体实施方式

概述

通常，对数据转换，很难部署依存包，尤其是从一个机器到另一个机器，主要是因为包所依赖的项目可能也可能不随一起移动。在此所述的系统和方法提供依存包的部署。在操作中，包组件可以使用枚举成功运行组件所需的所有文件的接口。也提供用于在部署期间合并组件和它们的依存关系的合并模块。在部署期间，包在合并模块缓存中搜索关联的合并模块。使用此信息，构建包括合并模块与该包依赖的组件的安装包为部署包，或安装文件。在实践中，安装包被发送到部署环境，以使得可以在部署环境中安装该包。

应理解，虽然在此所述的系统和方法是在依存包的部署的上下文中说明的，可以用超出所提供的例子范围的各种方法来使用部署技术。

A. 典型计算环境

图 1 展示适合的计算系统环境 100 的例子，在其中可以实现本发明。计算系统环境 100 仅仅是适合的计算环境的一个例子且并不对本发明的使用范围或功能造成任何限制。也不应将计算环境 100 解释为具有关联于在典型存在环境 100 中展示的任何一个组件或组件组合的任何依存关系或需求。

本发明可运行在大量的其他通用或专用计算系统环境或配置中。适合用于本发明的众所周知的计算系统环境和/或配置的例子包括但不仅限于，个人计算机、服务器计算机、手持或膝上设备、多处理器系统、基于微处理器的系统、机顶盒、可编程消费者电子设备、网络 PC、小型机、大型机、包含上述任何系统或设备的分布式计算环境，等等。

本发明可以在计算机可执行指令的通用的语境中说明，如由计算机执行的程序模块。通常，程序模块包括执行特殊任务或实现特殊抽象数据类型的例程、程序、对象、组件、数据结构等等。本发明也可以实现在分布式计算环境中，其中任务由通过通讯网络或其他数据传输媒体连接的远程处理设备执行。在分布式计算环境中，程序模块和其他数据可以位于本地和包含存储器存储设备的远程计算机存储媒体两者中。

参考图 1，用于实现本发明的典型系统包括形式为计算机 110 的通用计算设备。个人计算机 110 的组件包括但不仅限于处理单元 120、系统存储器 130，及连接包括系统存储器的各种系统组件到处理单元 120 的系统总线 121。系统

总线 121 可以为几种类型的总线结构中的任何一种，包括存储器总线或存储器控制器、外围设备总线，及使用多种总线结构中的任何一种的本地总线。作为例子，而非限制，这样的结果包括工业标准结构（ISA）总线、微通道结构（MCA）总线、扩展 ISA（EISA）总线、视频电子标准协会（VESA）本地总线及周边元素扩展接口（PCI）总线（亦称为包厢总线）。

计算机 110 通常包括各种计算机可读媒体。计算机可读媒体可以为任何能够由计算机 110 访问的可用媒体并包括易失和非易失媒体，可移动和不可移动媒体。作为例子，而非限制，计算机可读媒体包括计算机存储媒体和通讯媒体。计算机存储媒体包括以存储信息（如计算机可读指令、数据结构、程序模块或其他数据）的任何方法或技术实现的易失和非易失、可移动和不可移动媒体。计算机存储媒体包括，但不仅限于，RAM、ROM、EEPROM、闪存或其他存储器技术、CD-ROM、数字多用途盘（DVD）或其他光盘存储、盒式磁带、磁盘存储或其他磁存储设备，或可以用于存储所需信息且可以由计算机 110 访问的任何其他媒体。通讯媒体通常包括在如载波或其他传输机制这样的调制的数据信号中的计算机可读指令、数据结构、程序模块或其他数据并包括任何信息发送媒体。术语“调制的数据信号”指以这样的方式设置或改变它的一个或多个特征从而在该信号中编码信息的信号。作为例子，而非限制，通讯媒体包括有线媒体（如有线网络或直接有线连接）及无线媒体（如声音、射频、红外线和其他无线媒体）。上述媒体的任何组合亦应包括在计算机可读媒体的范围之内。

系统存储 130 包括形式为易失和/或非易失存储器如 ROM 131 和 RAM 132 的计算机存储媒体。包含基本的例程以帮助在个人计算机 110 的元素之间传输信息，如在启动过程中的基本输入/输出系统 133 (BIOS)存储在 ROM 131 中。RAM 132 通常包含可立即由处理单元 120 访问和/或操作的数据和/或程序模块。作为例子，而非限制，图 1 展示操作系统 134、应用程序 135、其他程序模块 136 和程序数据 137。

计算机 110 还可以包括其他可移动/不可移动、易失/非易失计算机存储媒体。仅作为例子，图 1 展示了读写不可移动、非易失磁媒体的硬盘驱动器 140、读写可移动、非易失磁盘的磁盘驱动器 151、读写如 CD-ROM 或其他光学媒体这样的可移动、非易失光盘 156 的光盘驱动器 155。也可以在例子操作环境中使用的可移动/不可移动、易失/非易失计算机可读媒体，如盒式磁带、闪存

卡、数字视频盘、数字录像带、固态 RAM、固态 ROM 等等。硬盘驱动器 141 通常通过不可移动存储器接口（如接口 140）连接到系统总线 121，而磁盘驱动器 151 和光盘驱动器 155 通常通过可移动存储器接口（如接口 150）连接到系统总线 121。

图 1 中展示的上述驱动器及其关联的计算机存储媒体对个人计算机 110 提供计算机可执行指令、数据结构、程序模块或其他数据的存储。例如，在图 1 中，硬盘驱动器 141 被示为存储操作系统 144、应用程序 145、其他程序模块 146 和程序数据 147。注意这些组件可以和操作系统 134、应用程序 135、其他程序模块 136 及程序数据 137 相同或不同。在此对操作系统 144、应用程序 145、其他程序模块 146 和程序数据 147 给出不同的编号以展示它们至少是不同的拷贝。用户可以通过如键盘 162 和定点设备 161（通常被称为鼠标、轨迹球或触摸垫）这样的输入设备输入命令和信息到个人计算机 110 中。其他输入设备（未标出）包括麦克风、操纵杆、游戏垫、圆盘式卫星电视天线、扫描仪或类似设备。这些和其他输入设备通常通过连接到系统总线的用户端口接口 160 连接到处理单元 120，但也可以通过其他接口，如并行口、游戏口，或通用串行总线（USB）来连接。显示设备 191 或其他显示设备也通过接口，如视频接口 190 与系统总线 121 连接。除显示器外，个人计算机通常包括其他外围输出设备（未标出），如扬声器 197 和打印机 196，它们可以通过输出外围设备接口 195 连接。

个人计算机 110 可以使用到一个或多个远程计算机（如远程计算机 180）的逻辑连接在连网环境中运行。远程计算机 180 可以是个人计算机、服务器、客户机、路由器、网络 PC、对等设备或其他普通网络节点，并且通常包括多个或所有上述相对于个人计算机 110 说明的元素，虽然在图 1 中只展示了存储器存储设备 181。所述逻辑连接包括局域网（LAN）171 和广域网（WAN）173，但也可以包括其他网络。这样的连网环境在办公室范围或企业范围的计算机网络、企业内部互联网和因特网中是很常见的。

当用在 LAN 连网环境中时，个人计算机 110 通过网络接口或适配器 170 与局域网 171 相连。当用在 WAN 连环境中时，个人计算机 110 通常包括调制解调器 172 或其他在广域网 173 上建立通讯的方法，如因特网。调制解调器 172，可以为内置的或外置的，通过用户输入端口接口 160 连接到系统总线 121。在连网环境中，相对于计算机 110 说明的程序模块或其部分，可以存储在远

程存储器存储设备中。作为例子，而非限制，图展示的远程应用程序 185 驻留在存储器设备 181 上。应理解所示的网络连接是说明性的且可以使用在计算机之间建立通讯连接的其他方法。

B. 典型联网计算环境

如上所述，计算机环境 100 可以作为计算机网络的部分来部署。总的来说，上面对计算机的说明适用于部署在网络环境中的服务器计算机和客户机计算机两者。图 2 展示典型网络环境，在其中可以实现本发明。如图 2 所示，多个服务器 10a、10b 等等通过通讯网络 14（它可以为 LAN、WAN、内联网、因特网或其他计算机网络）和多个客户机计算机 20a、20b、20c 或计算设备，如移动电话 15、固定电话 16 和个人数字助理 17 互连。例如，在通讯网络 160 为因特网的网络环境中，服务器 10 可以为 Web 服务器，客户机 20 通过任何数目的已知协议，如超文本传输协议（HTTP）或无线应用协议（WAP）和它通讯。每个客户机计算机 20 可以配备浏览器 180a 来获取对服务器 10 的访问。类似地，个人数字助理 17 可以配备有浏览器 180b 且移动电话 15 可以配备浏览器 180c 来显示并接收各种数据。

在操作中，用户（未示出）可以和运行在执行基于依存关系的包部署的客户机计算设备上的计算应用交互。报告可以存储在服务器计算机上并经通讯网络 14 通过客户机计算设备发送到合作的用户。

因此，本发明可以被用在包含用于访问并和网络交互的客户机计算设备 及用于和客户机计算机交互的服务器计算机的计算机网络环境的。然而，在此所述的系统和方法可以用各种基于网络的结构来实现，因此不应限制于所示例子。现在引用当前示意性的实施例更加详细地说明在此所述的系统和方法。

C. 包部署－基于依存关系－合并模块

基于依存关系的包部署提供组件依赖哪个文件的说明从而可以和包一起移动那些文件。在实践中，组件可以实现枚举该组件成功运行所需的所有文件的接口。在实践中，合并模块可以被用于合并用于部署的包内的依存关系文件和组件。特别地，合并模块允许创建可重用的安装组件。合并模块允许在合作的安装程序之间共享安装代码。

在一个特殊实施例中，合并模块（.msm 文件）是包含安装组件所需的文

件、资源、注册表项和安装逻辑的单个包。在操作中，合并模块通常不能被单独安装，但是相反可以被用在安装程序（.msi）文件的上下文中。合并模块允许获取特殊组件的所有依存关系，确保安装正确的版本。合并模块是强大的，因为一旦分发了合并模块，它就不能被修改；相反，对组件的每个后续版本创建新的合并模块。

比较起来，使用安装程序在另一个计算机上安装应用和文件；合并模块被用于安装由多个应用共享的组件。通常，两个类型之间的主要区别是目标受众。将由应用的最终用户直接使用的任何东西（可执行文件、文档、HTML页面）应被打包在安装程序中。只能由开发者使用的任何东西（DLL、控件、资源）被打包在合并模块中，然后开发者可以将它打包在安装程序中以分发给最终用户。再者，由多个应用共享（或可能由多个应用共享）的组件被放置在它自己的合并模块中。

图 3 为包部署的典型概念的方框图。如所示，第一个计算环境包括示意性的典型包 310。在包 310 内有任务 325、定制文件 315、平面文件 320、ftp 文件 330 和 DTS 文件 335。如虚线所示，包及其组件进行聚合然后如箭头所示发送到第二个计算环境。在所申请的发明中，包维护任务及其组件之间的依存关系的列表。将此信息作为部署的部分（如箭头所示）发送以使得该包在第二个计算环境中成功地执行和操作。

在例子中所提供的任务 325 可以包括用于计算、跟踪、关联、存储并发送公司销售的会计任务。相应地，任务 32 可能需要使用可以在平面文件 320 中找到的历史销售信息。另外，任务 325 可以使用 DTS.DLL 335 与其他包（如数据存储包）合作。

进一步来说，任务 325 可能需要计算销售额并转换销售额为各种国际货币。在此情况，任务 325 可能调用定制文件，如 Cust.DLL 315 来执行这样的操作。最后，任务 325 可能需要使用文件传输协议（ftp）（或其他通讯协议）发送该销售信息到第二个计算环境。在此情况，任务 325 调用 FTP.DLL 330 进行 ftp 类型的通讯。

如所示，执行它的很多功能的任务 325 依赖于其他文件、表和组件。亦如所示（如虚线框所示），包 310 的文件、表和组件并非创建为专门由任务 325 使用，而是可以在第一个计算环境中由其他组件、包、任务使用。当发生共享时，需要依存关系的正确表示来减少混淆并减少包创建中的错误。

应理解，虽然典型包据所述包含有特殊依存关系的特殊组件，这样的说明仅为示意性的且可以应用在此所述的创新的概念来创建有各种配置、组件和组件依存关系的各种包。

图 4 展示依存关系包部署系统 400 的方框图。如图所示，依存关系包部署系统 400 包括合并模块 435。合并模块与单元安装程序、包 430、配置文件 435、基于合并的表和文件 410 及文件 415 合作。通过在方框 405 处执行的对现有的基于合并的表和文件的扫描得到基于合并的表和文件 410 及文件 415。合并模块 420 与单元安装程序 420、包 430、配置文件 435、基于合并的表和文件 410，及文件 415 合作来产生依存的安装包及可执行安装文件 440，它们如箭头所示可部署到第二个环境。

在操作中，扫描现有的基于合并的表和文件以确定包 430 的组件是否有任何应加到包 430 中的依存关系。扫描产生基于合并的表和文件 410 及其他依存关系文件 415。基于合并的表和文件及其他依存关系文件的列表充当合并模块 425 的一个输入。合并模块 425 用于聚合所有的包组件及它们的依存文件和表并将它们与单元安装程序 420 及一些配置文件 430 合并来产生可执行安装文件 440。可执行安装文件 440，当在第二个环境中执行时，将提供包 330，包括它们所有的依存文件和组件。

在预期的实施例中，依存关系包部署系统 400 在创建安装文件和打包器列表 440 时使用 MICROSOFT® Visual Studio 的合并模块依存关系结构（如，合并模块及关联的文件）。特别地，在此情况，包 330 包括缺省的 DTS 组件。在方框 405 执行对现有的 MSM 表和文件的扫描来确定包 330 的非缺省（如第三方创建的）组件的文件和表依存关系。应理解，这样的处理是其中包任务暴露其依存关系且在其中任务能够向接口暴露这样的依存关系的处理的结果。在 Visual Studio 的情况下，所述接口是能够用于组装和部署包的包部署向导。所识别的 MSM 410 和其他文件 415 与单元 MSI 安装程序、包定义（如，缺省的包组件）和配置文件 435 由 Visual Studio 的标准合并模块合并来创建 MSI 列表和可执行安装文件 440。

图 5 展示在创建和部署包及其依存关系时执行的处理。如图所示，处理开始于方框 500 并进行至方框 505，其中包任务向合作的接口暴露其依存关系。从那里，处理进行至方框 510，其中暴露的依存关系被聚合。在方框 515，依存关系与用于部署的包缺省组件合并。然后包在方框 520 被部署且处理在方

框 525 终止。

图 6 展示在部署环境中部署包时执行的处理。如图所示，处理开始于方框 500 并进行至方框 503，在其中执行发送的安装文件（保护被部署的包及其依存关系）。然后在方框 510 展开安装文件中发现的包。然后在方框 515 部署包依存关系并在方框 520 执行包任务来校验正确的部署。从那里，处理在方框 525 终止。

D. 结论

总而言之，在此所示的系统和方法提供基于依存关系的包部署。然而，应理解，可以对本发明进行各种修改和替换结构。本发明并不限于在此所示的具体结构。相反，本发明试图覆盖所有的修改、替换结构和落在本发明范围和精神内的等价物。

应注意，本发明可以实现在各种计算机环境中（包括非无线及无线计算机环境）、部分的计算环境和现实世界的环境。在此所示的各种技术可以实现在硬件或软件或两者的组合中。较佳地，所述技术可以用每个都包括处理器、处理器可读的存储介质（包括易失和非易失存储器和/或存储元素）、至少一个输入设备和至少一个输出设备的可编程计算机上执行的计算机程序来实现。程序代码应用于使用输入设备输入的数据来执行上述功能并产生输出信息。输出信息应用于一个或多个输出设备。每个程序最好用高级过程或面向对象编程语言实现以和计算机系统通讯。然而，如果需要的话，所述程序可以用汇编或及其语言来实现。在任何一种情况，语言都可以为编译或解释语言。每个这样的计算机程序最后存储在由用于在存储媒体或设备由计算机读取来执行上述处理时配置和操作计算机的通用或专用可编程计算机可读存储媒体或设备（如 ROM 或磁盘）中。所述系统也可以被视为用计算机程序配置的计算机可读存储媒体来实现，其中这样配置的存储媒体使得计算机以特定的预定方式工作。

虽然在上面已详细说明了本发明的典型实施例，熟悉技术的人应即刻理解，可以在典型实施例中做出很多附加的修改而不实质性地偏离本发明的创新技术和优点。因此，这些和所有这样的修改应被包括在本发明的范围之内。本发明可以通过下面的权利要求更好地定义。

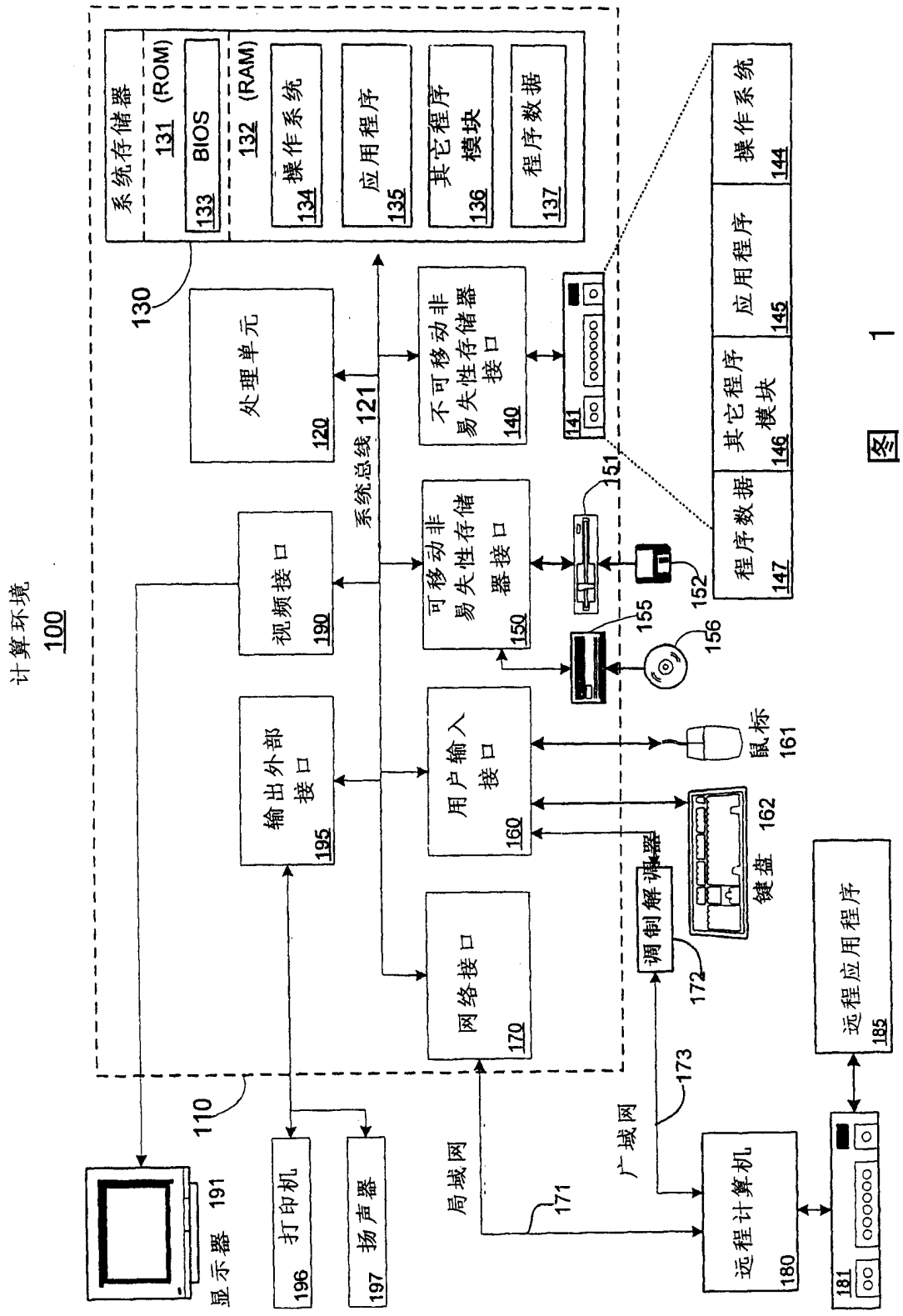

计算环境
100
系统存储器
131 (ROM)
133 BIOS
132 (RAM)
操作系统
134
应用程序
135
其它程序模块
136
程序数据
137
130
处理单元
120
系统总线 121
不可移动非易失性存储器接口
140
141
151
操作系统
144
应用程序
145
其它程序模块
146
程序数据
147
可移动非易失性存储器接口
150
152
155
156
视频接口
190
输出外部接口
195
用户输入接口
160
鼠标
161
键盘
162
网络接口
170
调制解调器
172
远程应用程序
185
110
显示器 191
打印机
196
扬声器
197
局域网
171
广域网
173
远程计算机
180
181

图 1

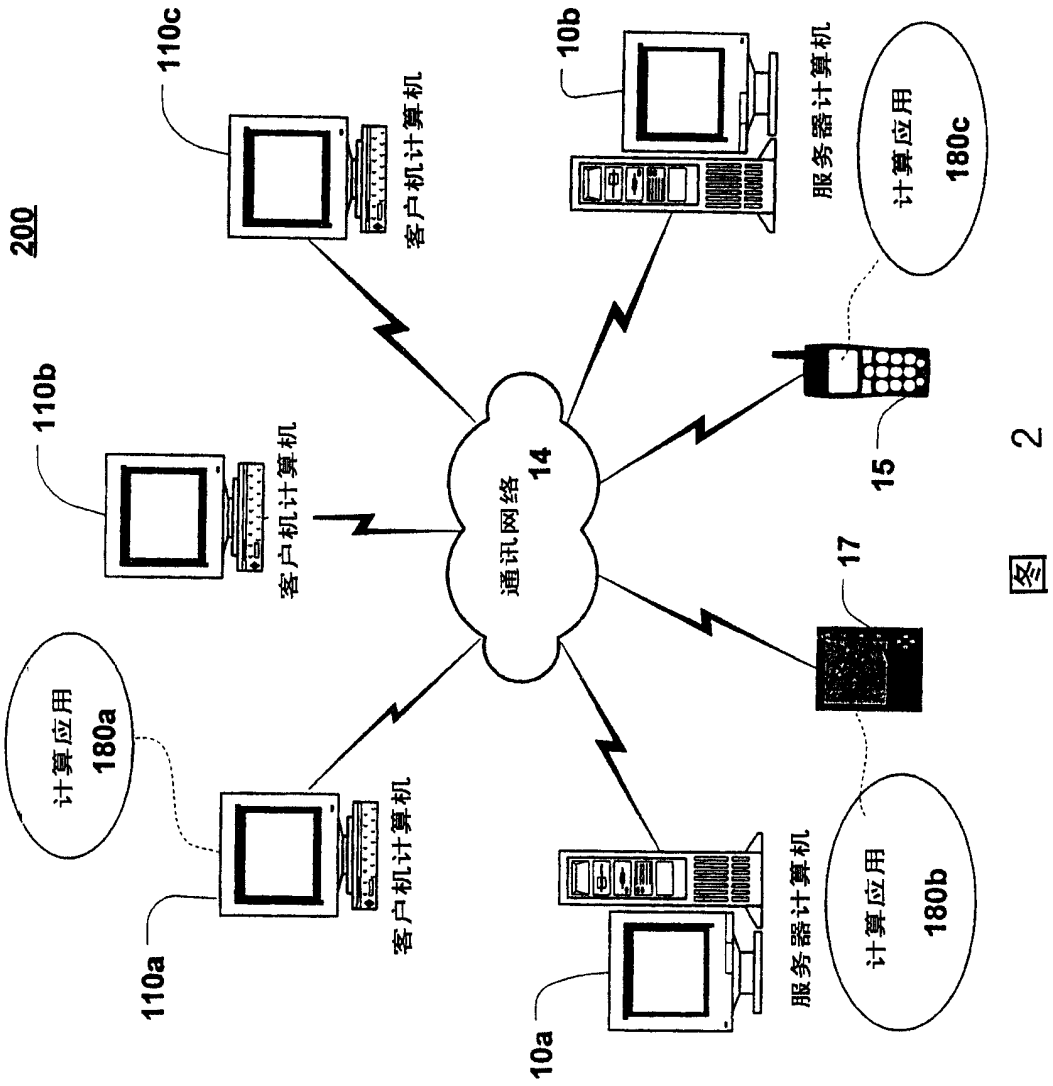

200
110c
客户机计算机
10b
服务器计算机
计算应用
180c
110b
客户机计算机
通讯网络
14
15
计算应用
180a
17
客户机计算机
110a
服务器计算机
计算应用
180b
10a

图 2

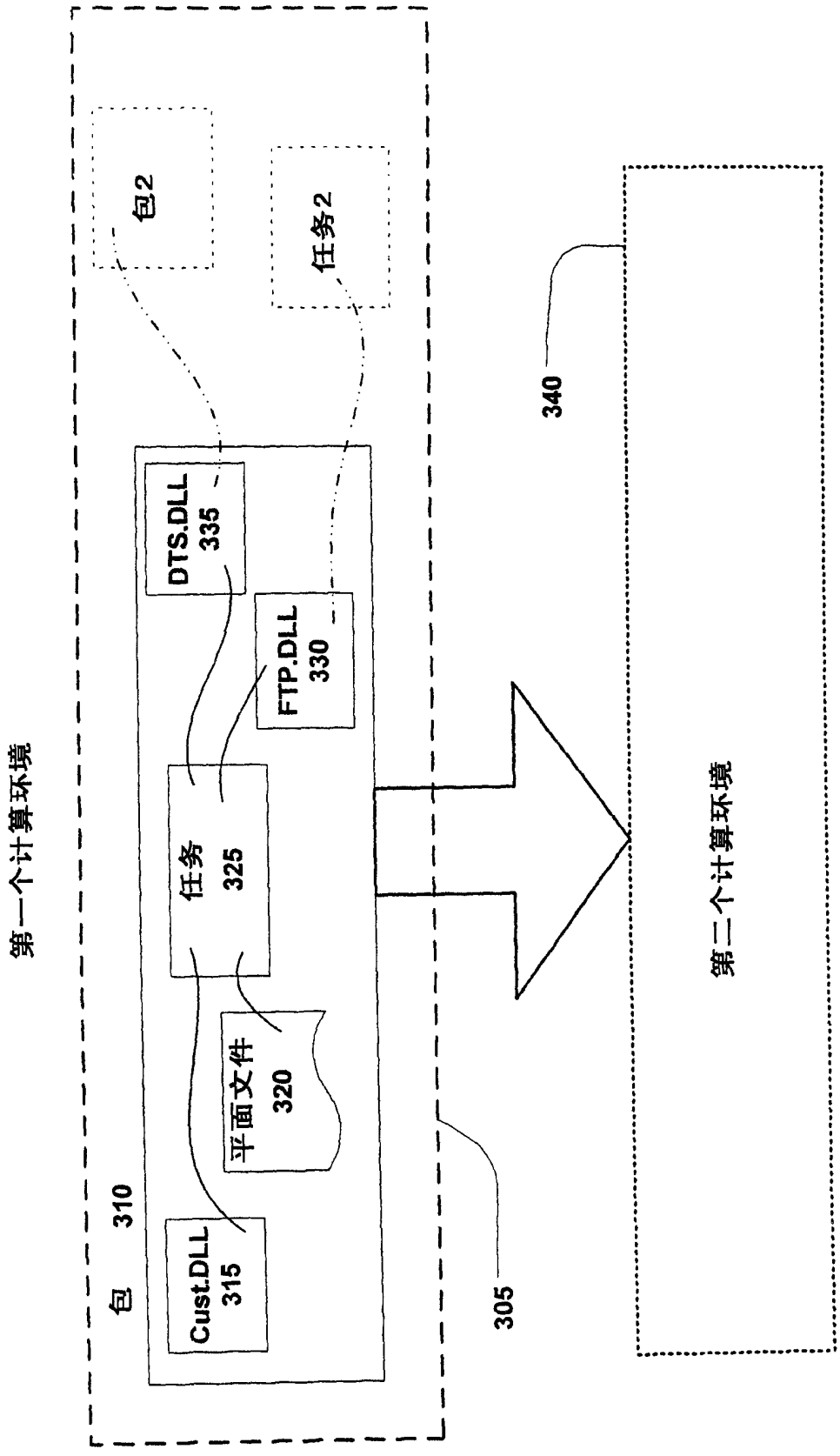

第一个计算环境
包2
任务2
DTS.DLL
335
FTP.DLL
330
任务
325
平面文件
320
Cust.DLL
315
包
310
305
340
第二个计算环境

图 3

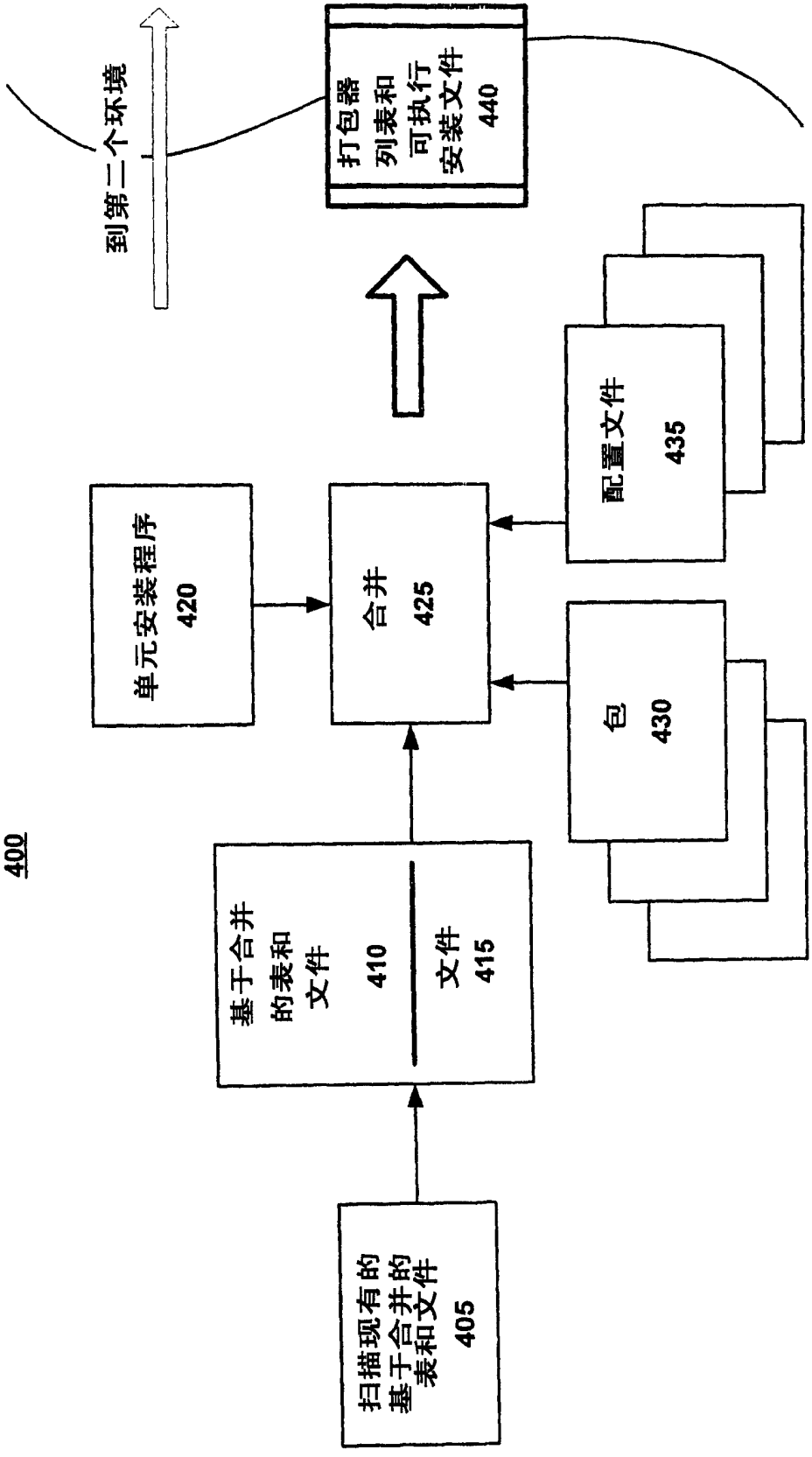

400
扫描现有的基于合并的表和文件 405
基于合并的表和文件 410
文件 415
单元安装程序 420
合并 425
包 430
配置文件 435
打包器列表和可执行安装文件 440
到第二个环境

图 4

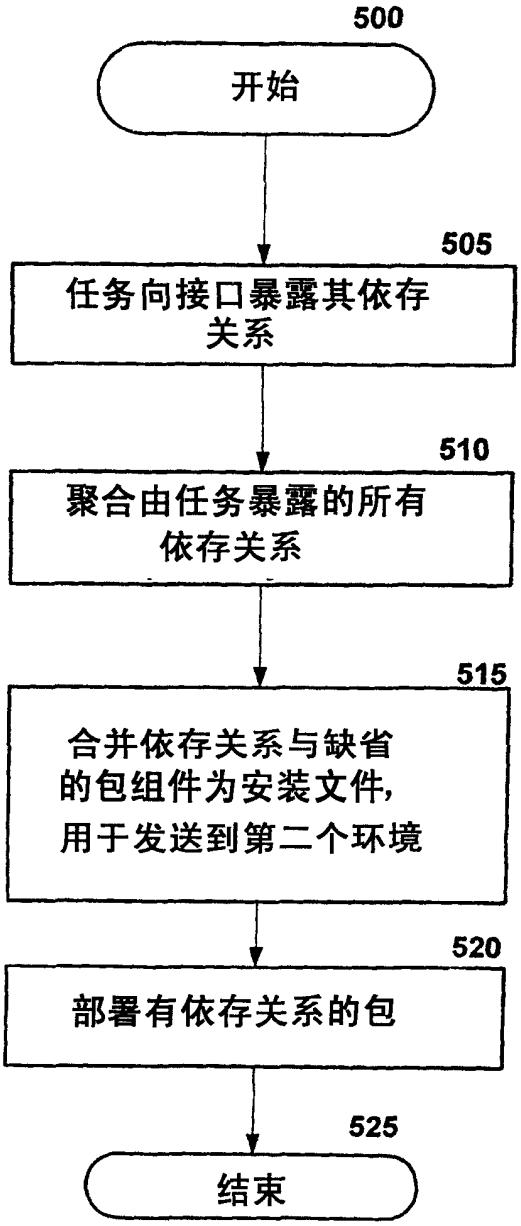

500
开始
505
任务向接口暴露其依存关系
510
聚合由任务暴露的所有依存关系
515
合并依存关系与缺省的包组件为安装文件，用于发送到第二个环境
520
部署有依存关系的包
525
结束

图 5

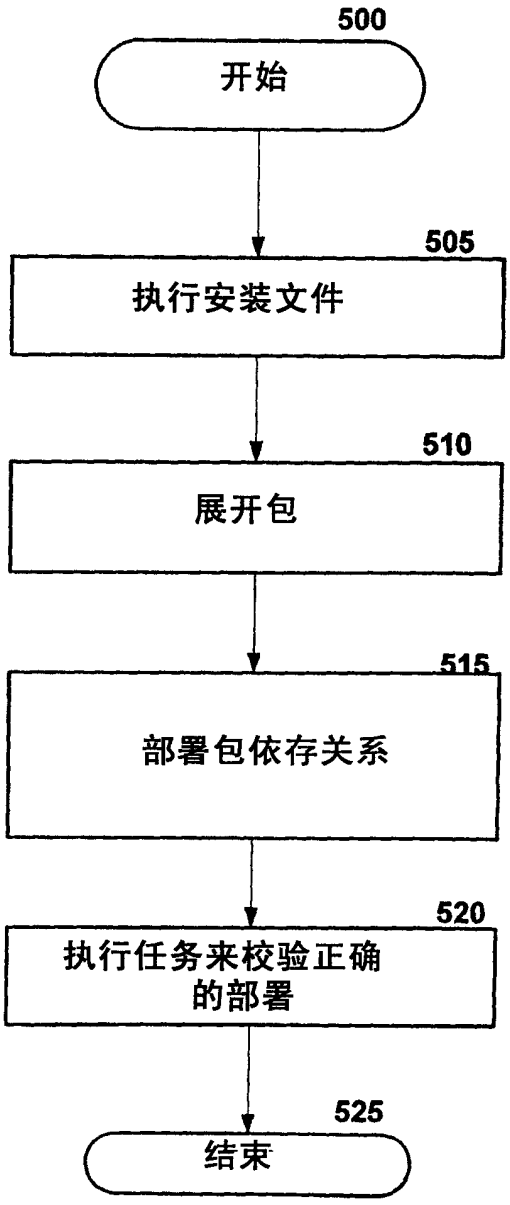

500
开始
505
执行安装文件
510
展开包
515
部署包依存关系
520
执行任务来校验正确的部署
525
结束

图 6