

[19] 中华人民共和国国家知识产权局

[51] Int. Cl.

G06F 17/30 (2006.01)



# [12] 发明专利说明书

专利号 ZL 200510009158.6

[45] 授权公告日 2009 年 8 月 19 日

[11] 授权公告号 CN 100530176C

[22] 申请日 2005.2.4

[21] 申请号 200510009158.6

[30] 优先权

[32] 2004.2.4 [33] US [31] 10/771,896

[73] 专利权人 微软公司

地址 美国华盛顿州

[72] 发明人 B·拉杰夫 G·霍尔

J·E·谢里弗尔

J·V·E·博尔杜西

M·弗斯特尔 P·拉玛尼 杨志东

[56] 参考文献

CN1310823A 2001.8.29

WO03/088613A1 2003.10.23

CN1398471A 2003.2.19

审查员 代 峰

[74] 专利代理机构 上海专利商标事务所有限公司

代理人 张政权

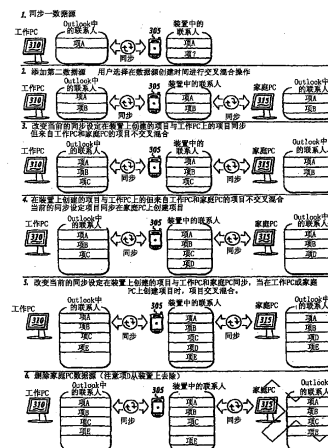
权利要求书 3 页 说明书 13 页 附图 15 页

[54] 发明名称

多种同步源的交叉混合

[57] 摘要

本发明使用户能让装置与可交叉混合的至少两个数据源同步。用户的装置被用以来回传送源之间的改变，并解决当在多个源上同时对项目作改变时的冲突。用户的装置追踪每个同步源具有何种项目版本，并使这些源的每一个同步为项目的最新版本。



1. 一种用来使一装置与数据源同步并允许所述数据源交叉混合的方法，其特征在于，包括：

创建一第一数据源和一第二数据源；

把所述装置连接到所述第一数据源；

使所述装置与所述第一数据源同步；

把所述装置连接到所述第二数据源；以及

使所述装置与所述第二数据源同步，其中所述装置被用作来回传送装置以在所述第一数据源和所述第二数据源之间交叉混合使得所述第一数据源、所述第二数据源和所述装置上均具有一项目的相同版本。

2. 如权利要求 1 所述的方法，其特征在于，其中使所述装置与所述第一数据源同步进一步包括将所述第一数据源上的一项目添加到所述装置中，并且在将该项目添加到所述装置之前，执行副本检测检查以确定该要被添加的项目是否已经被同步了。

3. 如权利要求 2 所述的方法，其特征在于，执行所述副本检测检查还包括在该要被添加的项目的属性和所述装置上现有项目的属性之间执行比较。

4. 如权利要求 2 所述的方法，其特征在于，执行所述副本检测检查还包括为该要被添加的项目计算同步散列值。

5. 如权利要求 2 所述的方法，其特征在于，还包括当该要被添加的项目已同步时更新该要被添加的项目。

6. 如权利要求 1 所述的方法，其特征在于，还包括在所述第一数据源和所述第二数据源之间交叉混合后，如果一项目不再被同步或已被物理地删除，则从在所述装置和数据源之间使用的同步协议接收删除该项目的删除命令并执行所述删除命令，其中所述删除命令从软删除和硬删除中进行选择。

7. 如权利要求 2 所述的方法，其特征在于，其中所述数据源之间的交叉混合能被用户限制。

8. 如权利要求 2 所述的方法，其特征在于，创建所述第一数据源和所述第二数据源还包括指示数据源类型，并存储关联于所述第一数据源和所述第二数据源中每个数据源的标识符。

9. 如权利要求 2 所述的方法，其特征在于，其中，使用第一同步协议使所述装置与所述第一数据源同步，而使用第二同步协议使所述装置与所述第二数据源同步。

10. 一种用于交叉混合数据源的方法，其特征在于，包括：

创建至少两个数据源以与一装置同步；

使所述装置与所述至少两个数据源同步；以及

通过所述装置被配置成在所述至少两个数据源之间用作来回传送装置以在所述至少两个数据源之间交叉混合数据使得所述至少两个数据源和所述装置上均具有一项目的相同版本。

11. 如权利要求 10 所述的方法，其特征在于，其中使所述装置与所述至少两个数据源同步进一步包括将一项目添加到所述装置中，并且在将该项目添加到所述装置之前，执行副本检测检查以确定该要被添加的项目是否已经被同步了。

12. 如权利要求 11 所述的方法，其特征在于，执行所述副本检测检查还包括为该要被添加的项目计算同步散列值。

13. 如权利要求 11 所述的方法，其特征在于，还包括在所述至少两个数据源之间交叉混合数据后，如果一项目不再被同步或已被物理地删除，则从在所述装置和所述至少两个数据源之间使用的同步协议接收删除该项目的删除命令并执行所述删除命令，其中所述删除命令从软删除和硬删除中进行选择。

14. 如权利要求 13 所述的方法，其特征在于，其中所述数据源之间的交叉混合能被用户限制。

15. 如权利要求 13 所述的方法，其特征在于，创建所述至少两个数据源还包括指示数据源类型，并存储关联于所述至少两个数据源中每个数据源的标识符。

16. 如权利要求 13 所述的方法，其特征在于，使用一种以上同步协议使所述装置与至少两个数据源同步。

17. 一种用于交叉混合数据源的系统，其特征在于，包括：

可相互交叉混合的至少两个数据源；

一装置，其被配置成在所述至少两个数据源之间担当来回传送装置以进行交叉混合，并被配置成与所述至少两个数据源同步，使得所述至少两个数据源和所述装置上均具有一项目的相同版本。

18. 如权利要求 17 所述的系统，其特征在于，其中使所述装置与所述至少两个数据源同步进一步包括将一项目添加到所述装置中，并且在将该项目添加到所述

装置之前，执行副本检测检查以确定该要被添加的项目是否已经被同步了。

19. 如权利要求 18 所述的系统，其特征在于，执行所述副本检测检查还包括为该要被添加的项目计算同步散列值。

20. 如权利要求 18 所述的系统，其特征在于，其中在所述至少两个数据源之间交叉混合数据后，如果一项目不再被同步或已被物理地删除，则在所述装置和所述至少两个数据源之间使用的同步协议接收删除该项目的删除命令，其中所述删除命令从软删除和硬删除中进行选择，所述装置被配置成处理软删除和硬删除命令。

21. 如权利要求 20 所述的系统，其特征在于，其中所述至少两个数据源之间的交叉混合能被用户限制。

22. 如权利要求 21 所述的系统，其特征在于，使用一种以上同步协议使所述装置与所述至少两个数据源同步。

---

多种同步源的交叉混合

## 技术领域

本发明涉及使一个装置与一个以上计算装置（“同步源”）同步。

## 背景技术

用户能使一个装置与另一个装置同步。例如，用户可使其工作计算机与其移动装置同步。同步之后移动装置和计算机具有被选中进行同步操作的相同信息。同步的基本目标是保持两个系统彼此保持流动。当设备同步时对一个源所作的改变可传播到另一个中。

例如，当用户将其移动装置上的联系人与其家庭计算机同步时，在完成同步操作后家庭 PC 和移动装置将有相同的联系人。许多用户将包括电子邮件、联系人、日程安排、以及文档等的项目同步。然而，要在两个以上不同装置间进行信息同步操作是非常困难的。所需要的是一种把多个同步源同步的方法。

## 发明内容

根据本发明的一方面，装置“交叉混合” (cross-pollinate) 其数据。一种诸如移动装置的计算装置追踪每个数据源具有何种项目版本，并将数据源同步为项目的最新版本。用户能可选择地允许或禁止对项目和数据源的交叉混合。

根据本发明的另一方面，副本检测算法帮助避免在装置间产生副本数据。

根据本发明的又一方面，取决于被删除项目与不再在用户过滤器中的项目相比较的差异，会对删除项目进行不同的处理。

根据本方面的再一方面，即使同步源使用了不同版本的同步协议，仍会发生交叉混合。

## 附图说明

图 1 和 2 示出的是可用于本发明一示例性实施例的示例性计算装置；

图 3 显示交叉混合同步系统；

图 4 示出了允许和禁止交叉混合；

图 5 示出的是示例性同步表格体系结构；  
图 6 示出了软删除和硬删除；  
图 7 示出了处理把项目添加到装置的命令；  
图 8 示出的是示例性副本检测场景；  
图 9 示出了处理用于装置上现有项目的副本添加命令；  
图 10 示出了 v1 和 v3 服务器的交叉混合；  
图 11 和 12 显示示例性同步场景；  
图 13 示出了数据源创建；  
图 14 示出的是用于去除数据源的过程；以及  
图 15 示出了根据本发明诸方面，使没有同步的数据源信息相一致。

### 具体实施方式

贯穿本说明书和权利要求，以下术语具有明确相关联的含义，除非上下文中另有明显指示。术语“交叉混合”指的是混和或混合来自多个同步数据源的数据。

### 说明性操作环境

参照图 1，用来实现本发明的一示例性系统包括诸如计算装置 100 的计算装置。在非常基本的配置中，计算装置 100 通常包括至少一个处理单元 102 和系统存储器 104。取决于计算装置的准确配置和类型，系统存储器 104 可以是易失的（诸如 RAM）、非易失的（诸如 ROM、闪存等等）、或两者的某种组合。系统存储器 104 通常包括操作系统 105、一个或多个应用程序 106，也可包括程序数据 107。在一实施例中，应用程序 106 可包括同步应用程序 120。该基本配置在图 1 中由虚线 108 内的那些组件示出。

计算装置 100 可具有附加特征或功能。例如，计算装置 100 还可包括诸如磁盘、光盘、或磁带的其它数据存储装置（可移动和/或不可移动）。这种附加存储器在图 1 由可移动存储器 109 和不可移动存储器 110 中示出。计算机存储介质可包括以用来存储诸如计算机可读指令、数据结构、程序模块、或其它数据的信息的任何方法或技术实现的易失和非易失、可移动和不可移动的介质。系统存储器 104、可移动存储器 109 和不可移动存储器 110 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、闪存或其它存储器技术、CD-ROM、数字化通用光盘（DVD）或其它光学存储器、磁卡、磁带、磁盘存储器或其它磁性存储装置、或可用来存储所需信息并可由计算装置 100 访问的任何其它介质。任

何这种计算机存储介质可以是装置 100 的一部分。计算装置 100 还可具有诸如键盘、鼠标、笔、语音输入装置、触摸输入装置等的输入装置 112。还可包括诸如显示器、扬声器、打印机等的输出装置 114。

计算装置 100 还可包含使装置能与其它计算装置 118 诸如在网络上进行通信的通信连接 116。通信连接 116 是通信介质的一个示例。通信介质通常体现为计算机可读指令、数据结构、程序模块、或其它诸如载波或其它传输机制的已调制数据信号，并包括任何信息传输介质。术语“调制数据信号”意指具有以这种把信息编码到信号中的方式来设置或改变的一个或多个特征的信号。作为示例，而非限制，通信介质包括诸如有线网络或直接有线连接的有线介质，以及诸如声学、RF、红外和其它无线介质的无线介质。在此所用的术语计算机可读介质包括存储介质和通信介质。

图 2 示出的是用于本发明一示例性实施例的移动计算装置。参照图 2，用来实现本发明的一示例性系统包括诸如移动计算装置 200 的移动计算装置。移动计算装置 200 包括处理器 260、存储器 262、显示器 228 和键盘 232。存储器 262 通常包括易失存储器（例如 RAM）和非易失存储器（例如 ROM、闪存等等）。移动计算装置 200 包括驻留于存储器 262 并在处理器 260 上执行的诸如微软公司 Windows CE 操作系统的操作系统 264 或另一操作系统。键盘 232 可以是按键式数字拨号盘（诸如典型的电话）、多键键盘（诸如常用键盘）。显示器 228 可以是液晶显示器，或其它通常用于移动计算装置的任何类型显示器。显示器 228 可以是触摸感应式的，且也可作为输入装置。

一个或多个应用程序 266 被装入存储器 262 并在操作系统 264 上运行。同步操作应用程序驻留于移动计算装置 200，并被编程以执行同步操作应用程序。同步操作应用程序可驻留于装置的硬件或软件中。移动计算装置 200 还可包括存储器 262 中的非易失存储器 268。可使用非易失存储器 268 来存储即使移动计算装置 200 断电也不应丢失的永久信息。

移动计算装置 200 包括可实现为一个或多个电池的电源 270。电源 270 还可包括诸如 AC 适配器或对电池供电或充电的充电器的外部电源。

如图所示移动计算装置 200 具有两种类型的任选外部通知机制：LED 240 和音频接口 274。这些装置可直接与电源 270 耦合，从而当被激活时，即使处理器 260 和其它组件都会关闭以节约电池电源，它们可保留由通知机制所指示的一段时间。音频接口 274 被用来向用户提供可听信号，并从用户处接收可听信号。例如，音频

接口 274 可与扬声器耦合用来提供可听输出，并可与话筒耦合用来接收可听输入，比如进行电话对话。

移动计算装置 200 还包括执行传送和接收通信功能的无线接口层 272。无线接口层 272 便于移动计算装置 200 与外部世界的无线连接。根据一实施例，在操作系统 264 的控制下进行无线接口层 272 的往返传送。换言之，可经操作系统 264 把由无线接口层 272 接收的通信传播到应用程序 266，反之亦然。

#### 多个同步源的交叉混合

图 3 根据本发明诸方面显示了交叉混合同步系统。用户可使诸如装置 1(305) 的计算装置与每个都被视为“同步源”的一个以上计算装置同步。

在一示例性交叉混合场景中，用户通常具有与两个同步源同步的单独装置。用户使该装置与两个源都同步，且交叉混合使源互相同步。当在多个源上同时对项目作了改变时，用户装置被用来在源之间来回运送改变并解决冲突。用户装置追踪每个同步源具有何种项目版本，并使源的每一个都同步为项目的最新版本。

作为交叉混合的示例，假设用户使其 PDA 装置 (305) 与家庭 PC (315) 以及工作 PC (310) 同步。一旦同步了，装置 (PDA 装置 305、工作 PC 310 以及家庭 PC 315) 的每一个都将包括项 A 320。当用户使其 PDA 同步于工作 PC 时，由 PDA 接收项 A。当用户使其 PDA 同步于家庭 PC 时，则从工作 PC 来到 PDA 的项 A 将自动传送到家庭 PC。另外，当允许交叉混合时，在家庭 PC 上创建的任何项目都会经 PDA 自动传送到工作 PC。

根据一实施例，用户可允许和禁止源之间的交叉混合。换言之，用户能在源之间交叉混合或使项目保持分离。该交叉混合设定能在任何时间开启和关闭。例如，用户可关闭与家庭 PC (315) 交叉混合的功能。当对家庭 PC 一禁止交叉混合时，来自工作 PC 的项目将不会移动到家庭 PC。另外，用户可使一个源保持分离，而另两个源交叉混合。这种灵活性帮助用户能对如何同步操作数据有完全的控制。

根据一实施例，当装置不再与单个源同步时，则从该装置清除所有数据。数据也可留于该装置上。当装置与两个或多个装置同步时，因为它与另一源同步数据被留在该装置上。对于未同步的数据源和已去除的那个数据源的追踪数据，ID 列被清除。

图 4 根据本发明诸方面示出了允许和禁止交叉混合。交叉混合同步装置知道它们正与之同步的源。根据一实施例，同步源配置信息存储在移动装置上并利用 AirSync 协议。该配置信息可存储在其它地方，且该装置可利用其它同步协议。



根据一实施例，管理员把策略设置在装置上以限制装置可与之同步的源的数量。通过把装置能同步的源限制成一预定量，可用该特征来阻止交叉混合。根据其它实施例，任何经授权的用户都可把策略设置在在装置上。

示例 1 显示了使一数据源（工作 PC 310）与 PDA 305 同步的同步操作。可见到项 A 在两个装置上复制。

示例 2 显示添加第二个数据源（家庭 PC 315）。PDA 305 与工作 PC 和家庭 PC 都同步，且每个装置包括在所有装置上的每个项目。原来在工作 PC 上的项 A，被复制到 PDA 305 和家庭 PC。原来在家庭 PC 上的项 B，被复制到 PDA 305 和工作 PC。

示例 3 示出了改变同步设定，使在装置上创建的项目与工作 PC 同步，且不允许工作 PC 和家庭 PC 交叉混合。可以见到，在装置上创建的项 C 与工作 PC 同步，但不对家庭 PC 交叉混合。

示例 4 显示在装置上创建的项目与工作 PC 同步，且在家庭 PC 上新创建的项目不对工作 PC 交叉混合。在此例中，项 D 在家庭 PC 上创建且与该装置同步，但不与工作 PC 同步。

示例 5 示出了改变同步设定，以跨越所有装置上交叉混合新创建的项目。在此例中，项 E 在工作 PC 上创建并与装置和家庭 PC 同步。根据一实施例，在不同同步设定中创建的项目保持那些同步设定。例如，如示例 3 和 4 中所示的在不同同步设定中创建的项 C 和项 D 是不交叉混合的。

示例 6 显示去除家庭 PC 同步源。根据一实施例，当那是该项唯一情况时，在被去除装置上创建的任何项目被去除。在此例中，从装置上去除了项 D，但由于项 B 也在工作 PC 同步，未去除项 B。

图 5 示出的是根据本发明诸方面的示例性同步表格体系结构。根据一实施例，SQL-CE 用于维护同步状态。

DB 文件（510）包含有对每种数据类型的单一表格以进行同步操作。该示例示出了同步的联系人、日程安排和任务。由于对一项目的改变在许多表格中发生，与该项目关联的 LastModified 字段（540）在每当编辑该项目时都得到了更新。这有助于确保通知并同步了所有的改变。项目表格、类别关联表格、或任何其它相关表格上对一个项目的改变触发对 LastModified 字段的更新。根据一实施例，对当前装置日期和时间设定值。在每次对项目的更新后接触该标记，有助于确保 SQL-CE 追踪系统正确追踪任何项目改变。如图所示，LastModified 列被添加到联系人

(524)、日程安排(520)和任务表格(522)中。每次改变项目时，都更新 LastModified 列。

以下是表格中该项目的示例性描述。

| 类别             | 常量               | 值 |
|----------------|------------------|---|
| DataSourceType | alAirSyncServer  | 1 |
|                | alAirSyncDesktop | 2 |

| 数据源 – 对每个数据源追踪的数据 |          |                     |
|-------------------|----------|---------------------|
| DataSourceID      | PK, GUID | 每个数据源的唯一标识符         |
| DataSourceType    | long     | 指示这是哪类数据源。用于追踪其能力。  |
| FriendlyName      | nvarchar | 数据源名，由应用程序值用来识别存储   |
| NetworkAddress    | nvarchar | 数据源地址，取决于数据源类型来设定值。 |

根据本发明的一实施例，将每种数据源的用户属性（同步用户属性 560）存储在登记表（Registry）中。这些用户属性包括同步引擎追踪的细节。根据一实施例，服务器追踪用户名、密码、区域、以及规则。桌面源追踪用户名、密码（任选的）、区域(domain)、规则以及机器名。

以下是根据本发明诸方面，示例性数据源应用程序接口的列表。

| 数据源 API                                                                                                                                                                                      |                                                                                                                                                                                                      |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 功能                                                                                                                                                                                           | 说明                                                                                                                                                                                                   |
| 创建数据源<br>HRESULT<br>CreateDataSource([in] UINT<br>cProps, [in] CEPROPVALS*<br>rgVals, [out] GUID<br>*pDataSourceID);                                                                         | 参数：<br>DataSourceType、FriendlyName、<br>NetworkAddress、ExchangeServer<br><br>创建新数据源，且如果成功则把新的<br>DataSourceID 传回 OUT 参数。<br><br>传递进来的 FriendlyName 是唯一的。如果<br>所请求的 FriendlyName 已经存在，会向调<br>用功能返回一错误。 |
| 编辑数据源<br><br>HRESULT<br>GetDataSourceProps([in]<br>GUID DataSourceID, [in]<br>UINT cProps, [in]<br>CEPROPVALS* rgVals);<br><br>HRESULT<br>SetDataSourceProps([in]<br>GUID DataSourceID, [in] | 把 DataSourceID 传递进来以指示要更新<br>的那个数据源。可编辑所有在创建操作中设<br>定的属性。<br>编辑这些字段的任一个都不需要复位同步<br>关键字。                                                                                                              |

|                                                                                                                                                                                                                                                                                                                                      |                                                                                                                                                                                                                                                                                               |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| UINT cProps,[in]<br>CEPROPVALS* rgVals);                                                                                                                                                                                                                                                                                             |                                                                                                                                                                                                                                                                                               |
| 删除数据源<br><br>HRESULT<br>DeleteDataSource([in] GUID<br>DataSourceID);                                                                                                                                                                                                                                                                 | 删除所请求的数据源, 并从表格中删除该数据源的列。<br><br>如果一项目曾与同步源同步, 但现在不再与其连接, 从表格中删除该项目。                                                                                                                                                                                                                          |
| 获取数据源<br><br>HRESULT<br>EnumDataSourceFirst([out]<br>HANDLE *phEnum, [out]<br>GUID *pDataSourceID);<br><br>HRESULT<br>EnumDataSourceNext([in]<br>HANDLE hEnum, [out]<br>GUID *pDataSourceID);                                                                                                                                        | 通过数据源列表进行检索和列举。                                                                                                                                                                                                                                                                               |
| 对项目设定来设定/检索<br>同步<br><br>struct SyncDataSource {<br>GUID DataSourceID;<br>BOOL fSyncing;<br>}<br><br>HRESULT<br>GetDataSourceForItem([in]<br>GUID itemID, [in] GUID<br>itemType, [out]<br>SyncDataSource **ppSDS);<br><br>HRESULT<br>SetDataSourceForItem([in]<br>GUID itemID, [in] GUID<br>itemType, [in]<br>SyncDataSource *pSDS); | GetDataSourceForItem<br>对于新项目, 调用该 API 以确定该 itemType 可与之同步的数据源列表以及缺省值(新项目应当用缺省与之同步的那个数据源)。对于现有项目, 调用该 API 以确定该项目可与之同步的有效数据源列表以及当前同步值 (项目当前作同步操作时)<br><br>使用该 API 对已指定 itemID 的值同步设定<br><br>SetDataSourceForItem<br>该功能响应于若干情况:<br>1: 新项目-设置未决标记<br>2: 用相同值编辑 - 不动作<br>3: 用新值编辑- 适当时标记为未决删除或未决 |

#### 硬删除和软删除之间的差异

图 6 根据本发明诸方面示出了软删除和硬删除。根据一实施例, 如果项目从过滤器上消失 (不再同步) 或已被物理删除了, 同步协议发送对项目的删除。根据一实施例, 用于对项目交叉混合的过滤器设定是装置范围的设定。由于因在交叉混合源上过滤器外删除的传播而可能有数据丢失, 使用硬删除命令有助于区别软删除

和硬删除。

术语“硬删除”指的是从每个存储中物理地删除该项目。硬删除跨越所有交叉混合源传播。术语“软删除”指的是项目已在过滤器上消失的事实。软删除从该装置中删除该项目，但并不把该删除传播到任何其它源。

如果同步协议的一个版本不支持软删除，则当接收删除命令时，该装置仅从其存储中删除该项目。换言之，删除并不交叉混合。因为有可能从上次同步操作起项目可在过滤器中消失，软删除发生在同步进程的末期。

例如，从服务器中删除项目 610，则会在该装置上删除该项目而不会在家庭 PC 上删除该项目，如删除 620 所示。另一方面。如果在工作 PC 上硬删除该项目，则项目 610 在装置和家庭 PC 上都被删除，如删除 625 所示。当装置接收到对不存在项目的删除命令时，忽略删除命令。

#### 副本检测

图 7 示出的是根据本发明诸方面，处理把项目添加到装置中的命令。在把项目添加到装置中之前，要完成副本检测检查。执行副本检测检查有助于确保正在进行交叉混合的不同装置上未创建副本。根据一实施例，副本检测基于项目属性比较，且副本检测代码在装置上实现（诸如图 3 中所示的装置 305）。

在方框 710，装置从被同步的数据源接收添加命令。移到方框 715，装置为要添加的项目计算散列（hash）值，并对存储搜索具有相同散列值的任何其它项目。散列值被用来唯一地标识项目。也可使用其它唯一标识项目的方法。移到判定方框 720，确定装置是否交叉混合。

如果在判定方框 725 散列被发现，装置将视该项目为副本，并移向判定方框 735。当未发现同步散列时，过程移向方框 730，在那里把项目添加到该装置中并设置同步未决位。在判定方框 735，如果被发现项目当前与该数据源同步，则在方框 730 创建该项目，并设置同步未决字段。根据一实施例，如果装置上带有匹配同步散列的项目已与该源同步时，装置将不考虑把任何项目视为副本。

当项目当前不与数据源同步，则过程移到方框 740，在那里该项目被检测为副本。此时，装置将保留已有版本并仅把新项目 ID 添加到现有记录中。

图 8 根据本发明诸方面，示出了示例性副本检测场景。通常，当把项目添加到装置上时，在项目属性的子集上计算 SyncHash（同步散列 810）。项 A 具有值为 HASH 101(815)的同步值。当装置接收要添加的项目时，装置比较新项目 and 现有项目的 SyncHash，使装置能够检测副本。

参照步骤 1，来自服务器 1 的项 A 已与装置同步。此时，项 A 在服务器 1 和带有 HASH 101 的 SyncHash 值的装置上。在步骤 2，用户把项 A 从其装置拷贝到服务器 1 上，从而在服务器 1 上创建项 B。项 B 有与项 A 相同的 SyncHash 值。当在步骤 3 装置与服务器同步时，将在装置上创建项 B，因为它已与服务器同步。

计算散列

在计算同步散列时采用了两种级别的属性级别匹配。

准确数据比较（主要关键字组）：一组字段被定义为主要属性，把属性进行比较以把该项目视为副本。该组字段对每种数据类型而言可以是不同的。选定的该组字段应当不经重大代码改变就能适应代码。这可包括将这些字段列在标题文件中，从而在测试过程中可调节算法。

属性存在（次要关键字组）：几乎包罗所有的更大的一组字段，用来检查这些属性中数据的简单存在。这有助于确保，如果在一项目中某字段被设定而在另一项目为空白，则即使其主要关键字匹配，项目也不会被视为彼此的副本。

存储/更新散列

当首次同步项目时，计算 SyncHash 并将其存储在该项目上。当对已同步的项目进行编辑时，更新 SyncHash 值。以下是根据本发明的一实施例，用于联系人、日程安排和任务的示例性主要和次要关键字组。其它属性被用来为要同步的其它项目创建主要和次要关键字组。

|                    |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|--------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 联系人同步散列算法          | 属性                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| 主要关键字组<br>(准确数据比较) | FirstName<br>LastName<br>HomePhoneNumber 的最后 4 个字符<br>BusinessPhoneNumber 的最后 4 个字符<br>(如果少于 4 位数使用)                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| 次要关键字组<br>(属性存在)   | Anniversary, AssistantName, AssistnamePhoneNumber, Birthday, Body, Business2PhoneNumber, BusinessCity, BusinessCountry, BusinessPostalCode, BusinessState<br>BusinessStreet, BusinessFaxNumber, CarPhoneNumber, CompanyName, Department, Email1Address, Email2Address, Email3Address, FileAs, Home2PhoneNumber, HomeCity, HomeCountry, HomePostalCode, HomeState, HomeStreetm<br>HomeFaxNumber, JobTitle, MiddleName, MobilePhoneNumber, OfficeLocation, OtherCity, OtherCountry, OtherPostalCode<br>OtherState, OtherStreet, PagerNumber, RadioPhoneNumber, Spouse, Suffix, Title, WebPage, YomiCompanyName, YomiFirstName, YomiLastName |
| 日程安排同步散列算法         | 属性                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |

|                    |                                                                                                                                                                                                                                  |
|--------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 主要关键字组<br>(准确数据比较) | UID, StartTime, Location, Subject                                                                                                                                                                                                |
| 次要关键字组<br>(属性存在)   | AllDayEvent, Email, Name, DtStamp, EndTime, Deleted, ExceptionStartTime, MeetingStatus, OrganizerEmail, OrganizerName, Recurrence, Type, Until, Occurrences, Interval, DayOfWeek, DayOfMonth, WeekOfMonth, MonthOfYear, Reminder |

|                    |                                                                                                             |
|--------------------|-------------------------------------------------------------------------------------------------------------|
| 任务同步散列算法           | 属性                                                                                                          |
| 主要关键字组<br>(准确数据比较) | Subject, DueDate, Categories                                                                                |
| 次要关键字组<br>(属性存在)   | StartDate, CompletedDate, Importance, IsComplete, IsRecurring, Sensitivity, ReminderSet, ReminderTime, Body |

如果项目被标记为交叉混合，装置上提供的数据将与其它数据源进行交叉混合。项目的截断版由于装置限制可交叉混合。根据一实施例，当发生截断时，把文本（Message Truncated）（消息被截断）添加到 Notes（备注）字段的最后。用户将因此知道他们没有完全的原始拷贝。

装置上的同步引擎首先从数据源请求所有的添加。然后它把所有仍存在的添加（被复制添加作了标记，因此不会被发出）发送到数据源。

根据一实施例，装置上的项目追踪三个属性。ServerID 和 SyncHash 将作为每个项目上的用户属性被存储。

| 属性            | 说明                                                        |
|---------------|-----------------------------------------------------------|
| ItemID        | 用于标识该项目。                                                  |
| ServerID1...N | 这是在服务器/台式机上标识项目的 ID<br>该项目有多个 Server ID 在项目上作为项目中用户属性的存储。 |
| SyncHash      | 这是用以检测副本的同步散列值。在每次改变中都计算/重新计算该值。                          |

ServerID 属性有四个状态：

| 同步状态        | 说明                                                    |
|-------------|-------------------------------------------------------|
| 同步未决        | 项目在装置上且还未与目标数据源同步。在下次同步时，在数据源上创建该项目。                  |
| 有效 ServerID | 存储项目 ServerID 使与数据源的正确映射之用                            |
| 未决删除        | 在过去该项目对该数据源已同步，但用户已请求它不再同步。在下次同步操作时，从数据源删除该项目，并清除该属性。 |
| 未同步         | 项目未与该数据源同步                                            |

图 9 显示根据本发明诸方面，处理用于复制装置上现有项目副本的添加命令。

对于被视为副本的项目，过程移到方框 910，在那里添加命令被转换成改变命令。不是把整个项目拷贝到装置中，而是把项目更新。在方框 920，属性列表与已知模式相比较。移到方框 930，对于未提供的已知属性，显式删除被添加到改变命令中。方框 940 示出了经更新命令。在方框 950，处理改变命令。

#### 不同同步版本间的交叉混合

实施属性级别改变追踪是为了帮助支持多个同步协议版本交叉混合，且使不必要的写动作最少。每个同步提供者负责获知其同步的属性组，并仅当其关心的属性中发生改变时启动同步。由于同步供应者仅改变知道的属性，有可能装置上的项目将包含合并的属性组，部分属性来自 v1 服务器而其它属性来自同步的 v3 服务器。

图 10 示出了根据本发明诸实施例，v1 和 v3 服务器的交叉混合。同步协议的不同版本支持不同属性。如实例中所示，v1 协议支持属性 A 和 B，而 v3 协议支持属性 A、B 和 Z。

开始，装置未与 v1 服务器或 v3 服务器同步。在 910，装置首先与 v1 服务器同步，并接收项 A 和 B。然后，在 920，装置与 v3 服务器同步。项 A 和 B 被检测为副本，且项 A、B 和 Z 与装置同步。在 930，当装置再次与 v1 服务器同步时，没有同步操作发生，因为仅有项目 Z 作了改变而它不被 v1 服务器支持。

图 11 显示了根据本发明诸方面的示例性同步场景。步骤 1110 显示项 A 先前已与家庭 PC 及工作服务器交叉混合。在 1120，编辑项 A 来创建项 A 的版本 2(v2)。在 1130 装置与工作服务器同步，其中版本 2 与该装置同步。然后，在方框 1140 装置与家庭 PC 同步。由于装置被设定为服务器 Wins，现在项 A（版本 1）在装置上。当装置再次与工作服务器同步时无冲突，而现在项 A 的版本 1 在工作服务器上（1150）。

图 12 根据本发明诸实施例显示示例性同步场景。步骤 1210 显示项 A 先前与家庭 PC 和工作服务器交叉混合。在 1220，标识了工作服务器上的同步关键值为零。在方框 1230 用户进行交叉混合操作从而工作服务器 ID 被删除。移到方框 1240，同步从头开始，且装置接收到请求把项 A 添加到装置中的命令。项目被检测为副本，且保留了来自家庭 PC 的项目。工作服务器 ID 被添加到项目标记中。

#### 创建数据源

图 13 示出了根据本发明诸方面的数据源创建。开始框之后，过程移到方框 1310，在那里获取新数据源的账户信息。移到方框 1320，作了把新数据源登记到系统中的请求。在方框 1330，数据源类型被设置为关联于数据源的类型。根据一

实施例，数据源类型与装置类型（即 DesktopSync 和 ServerSync）相关。然后，在方框 1340，记录了机器名。如果丢失了 DesktopSync 信息，这促使装置有能力与台式机（大部分情形）重新连接。然后过程移动到结束方框，并返回以处理其它动作。

#### 去除数据源

图 14 显示了根据本发明诸方面用于去除数据源的过程。删除数据源的动作删除了数据源以及与该数据源同步的所有项目。如果项目与一个以上数据源同步或未决同步，则不删除该项目。

在开始方框后，过程移到方框 1410，在那里确定有多少源与装置同步。移到判定方框 1420，当仅有一个数据源时，过程移到方框 1430，在那里从装置中去除与被删除源同步的项目。当有一个以上装置时，过程移到方框 1440，在那里去除装置上未交叉混合项目。转换到方框 1450，去除已删除源的 ID。根据一实施例，从应用表格中去除该数据源的用户列。根据一实施例，在去除任何数据之前，向用户提供警告消息。

以下是作进一步阐述的示例。假设，装置使其联系人和日程安排与第一源同步，而使日程安排和任务与第二源同步。然后用户删除第一源。作为去除该第一源的结果，从装置上清除所有的联系人。日程安排项目和任务留在装置上，且第一源的 ID 被去除。

#### 协调同步之外的数据源信息

图 15 示出了根据本发明诸方面，协调没有同步的数据源信息。

使用以下示例进行说明。工作 PC 不知道的数据源（工作 PC）在装置上存在。这会因很多原因而发生。例如，如果用户不得不重新安装同步程序或重建机器，它会发生。当装置连接到台式机上时，台式机通知装置已有与其同步的数据源设置（1510）。这可通过比较装置上数据源的网络机器名和工作 PC 名来确定。删除现有的台式机源，并重建一个新源。

装置不知道的数据源在台式机上存在。这可在比如装置冷启动时发生。通过比较存储在台式机上的数据源的 DeviceID 和装置的 DeviceID，可确定装置过去与台式机同步，且可确定是否应再次设置装置使其与台式机同步。如果用户说“是”，则为用户创建新数据源。

以上说明书、示例和数据提供了本发明组成部分的制造和使用的完整描述。因为可作本发明的许多实施例而不背离本发明的精神和范围，本发明归属于所附权



利要求。

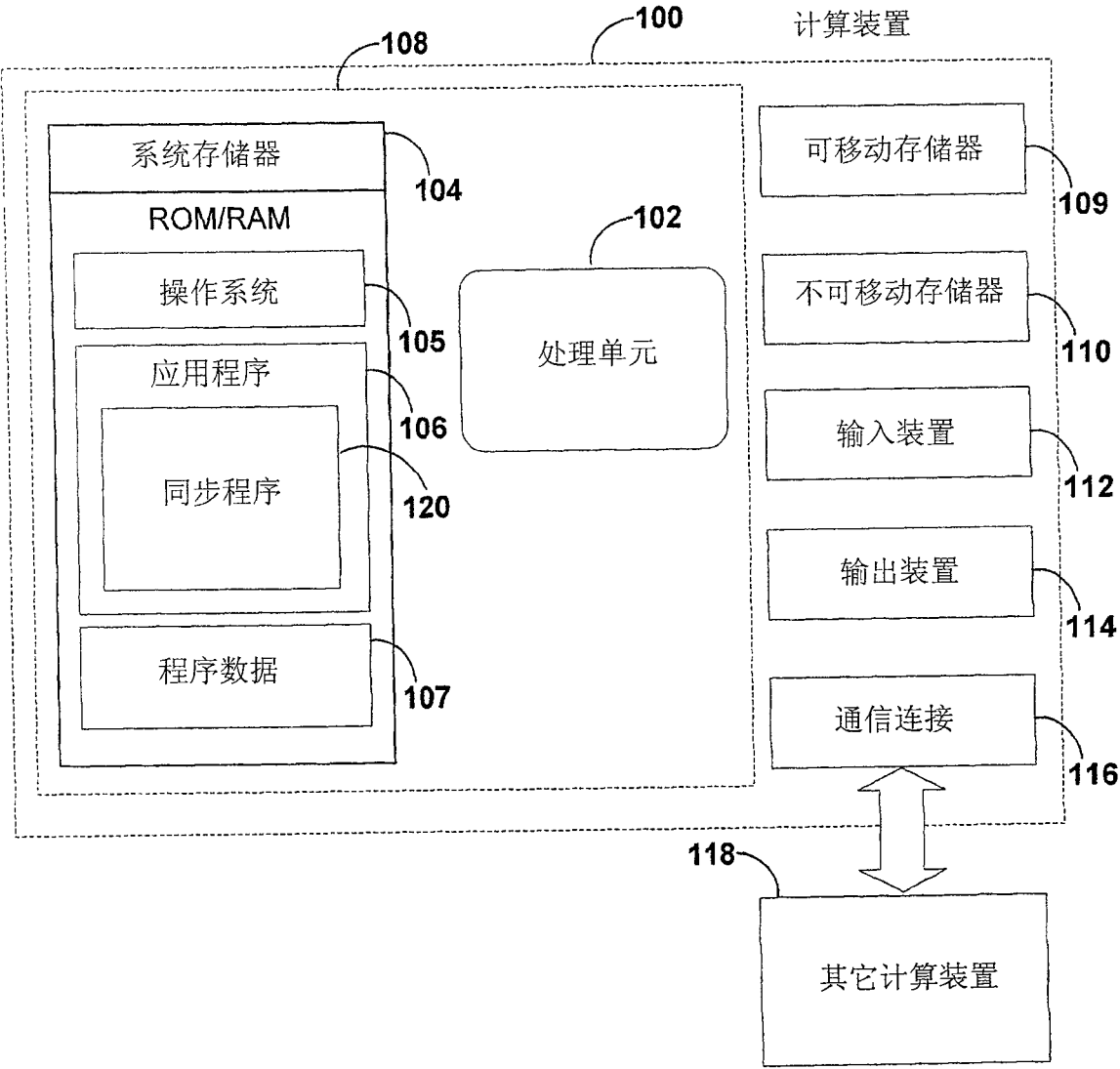


图 1

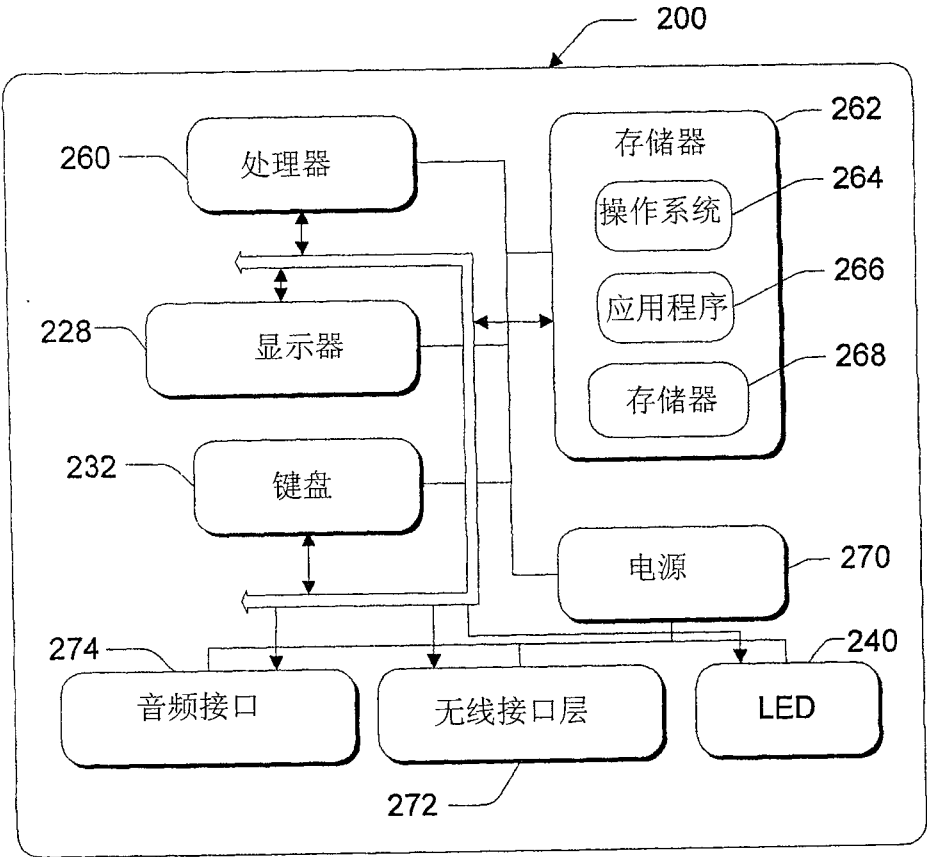


图 2

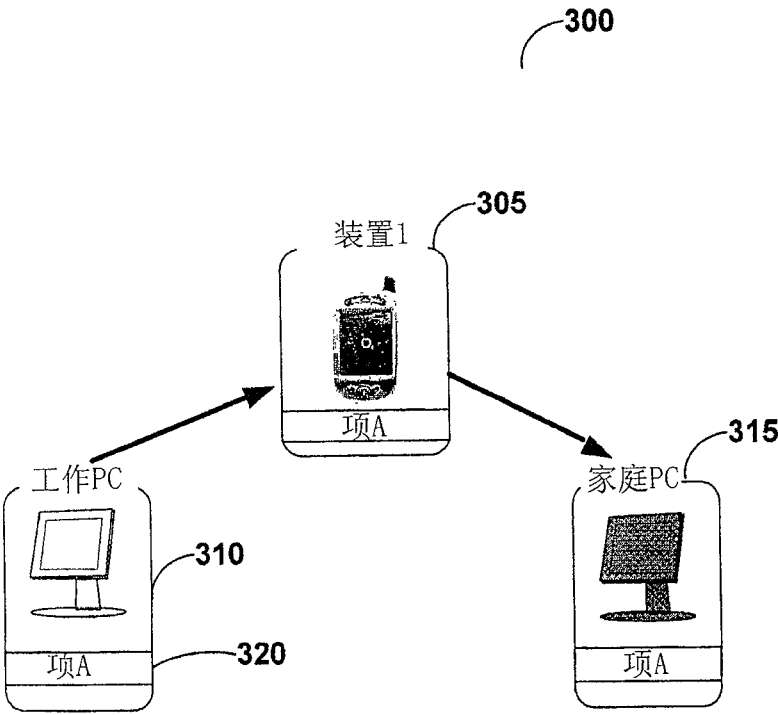


图 3

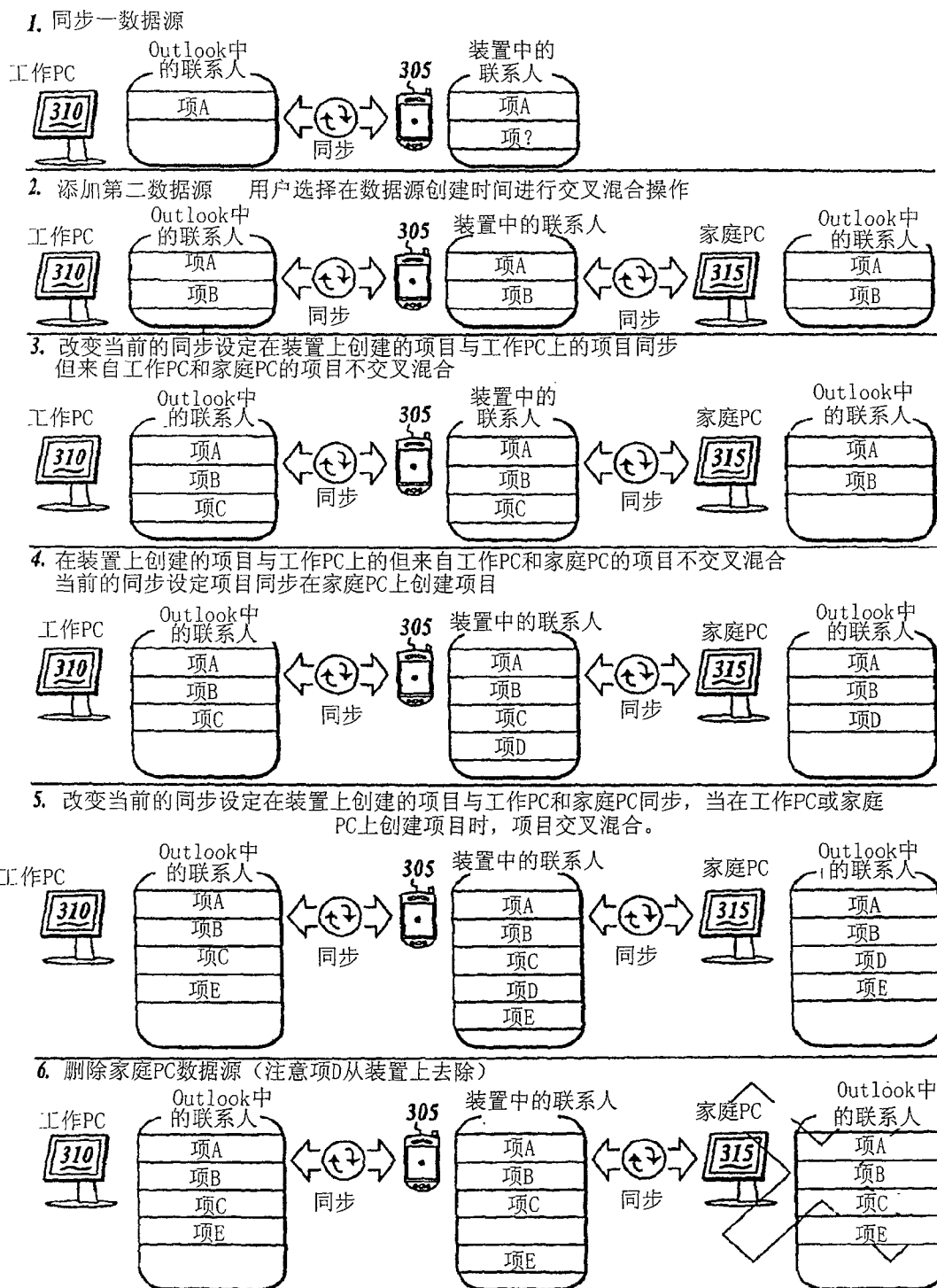
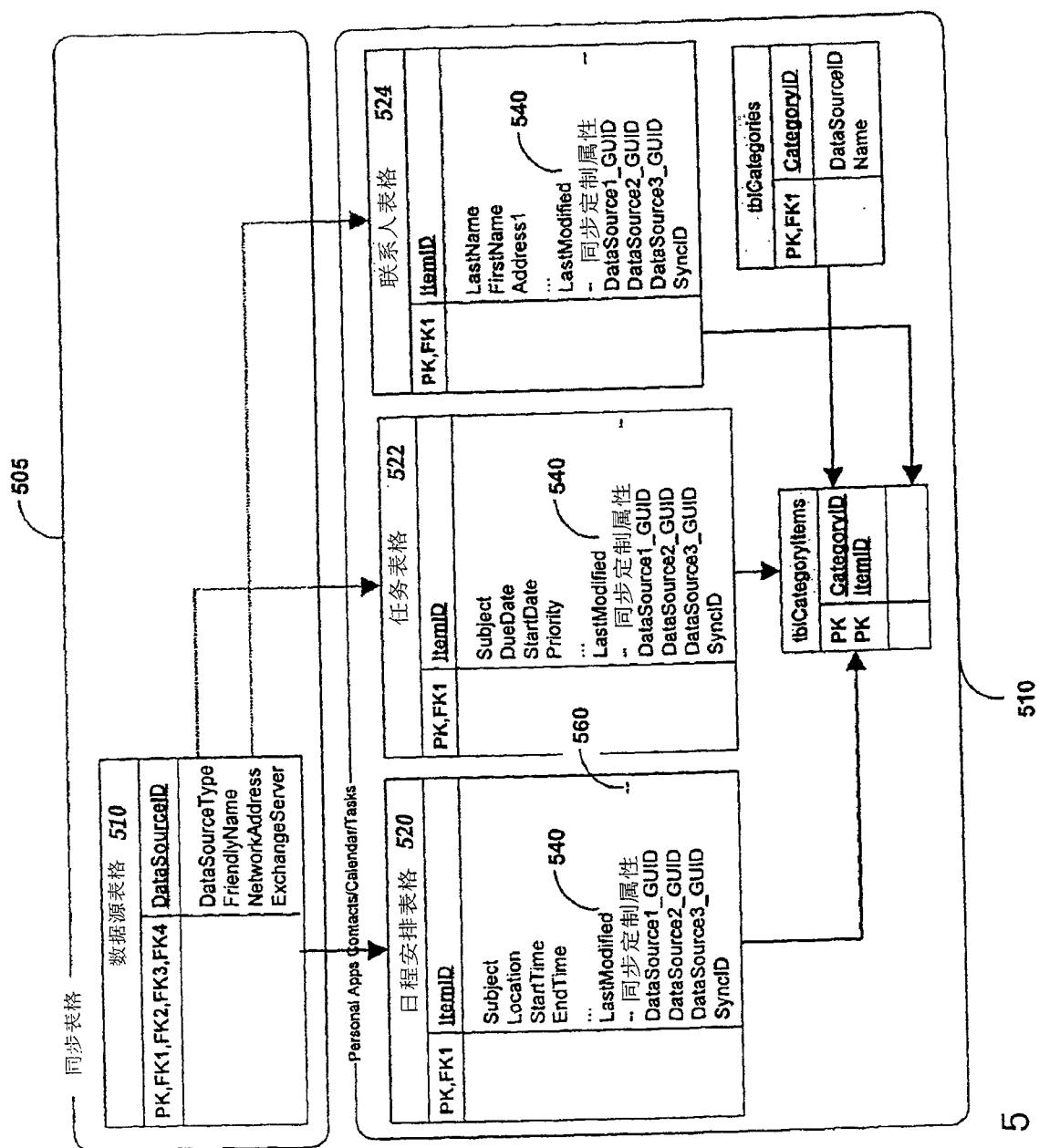


图 4



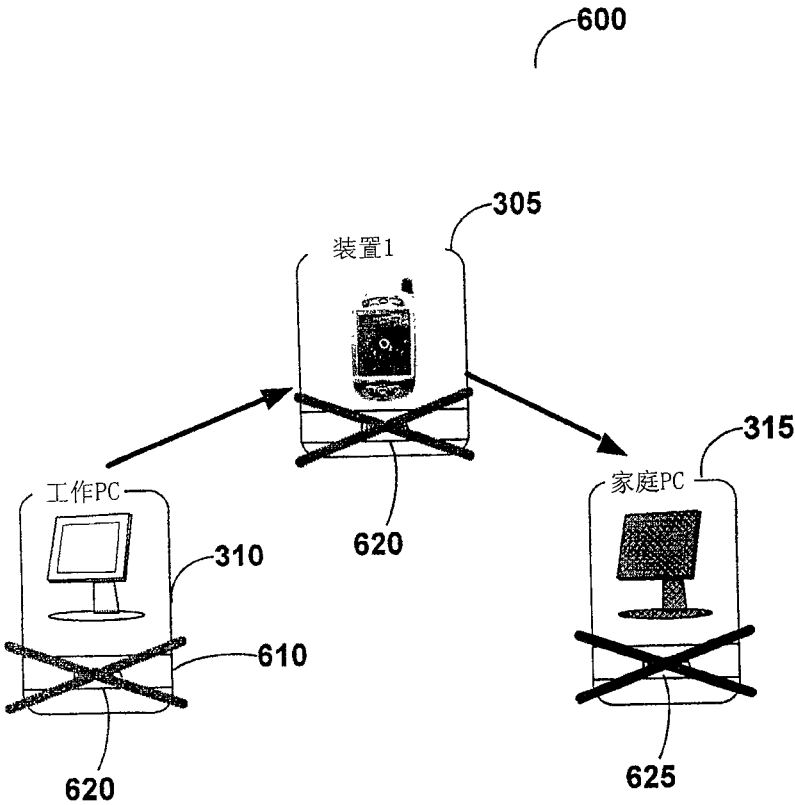


图 6

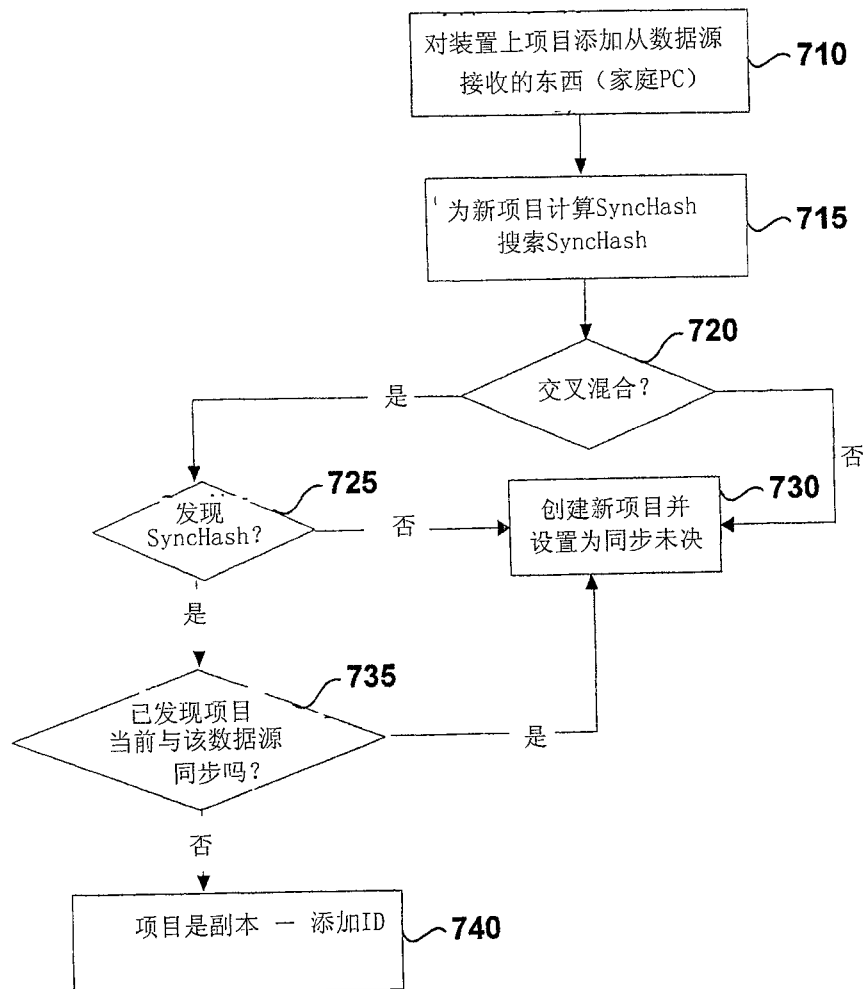
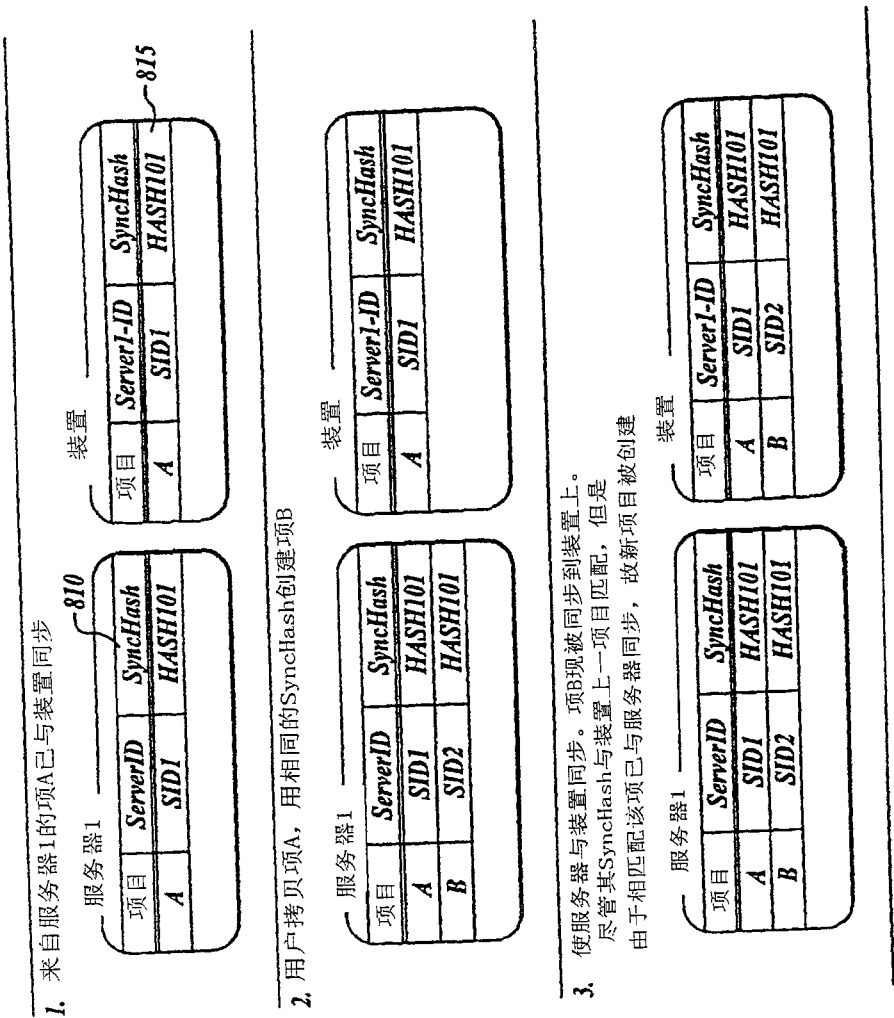


图 7





服务器1

| 项目 | ServerID | SyncHash |
|----|----------|----------|
| A  | SID1     | HASH101  |
| B  | SID2     | HASH101  |

装置

| 项目 | Server1-ID | SyncHash |
|----|------------|----------|
| A  | SID1       | HASH101  |

服务器1

| 项目 | ServerID | SyncHash |
|----|----------|----------|
| A  | SID1     | HASH101  |
| B  | SID2     | HASH101  |

装置

| 项目 | Server1-ID | SyncHash |
|----|------------|----------|
| A  | SID1       | HASH101  |
| B  | SID2       | HASH101  |

图 8

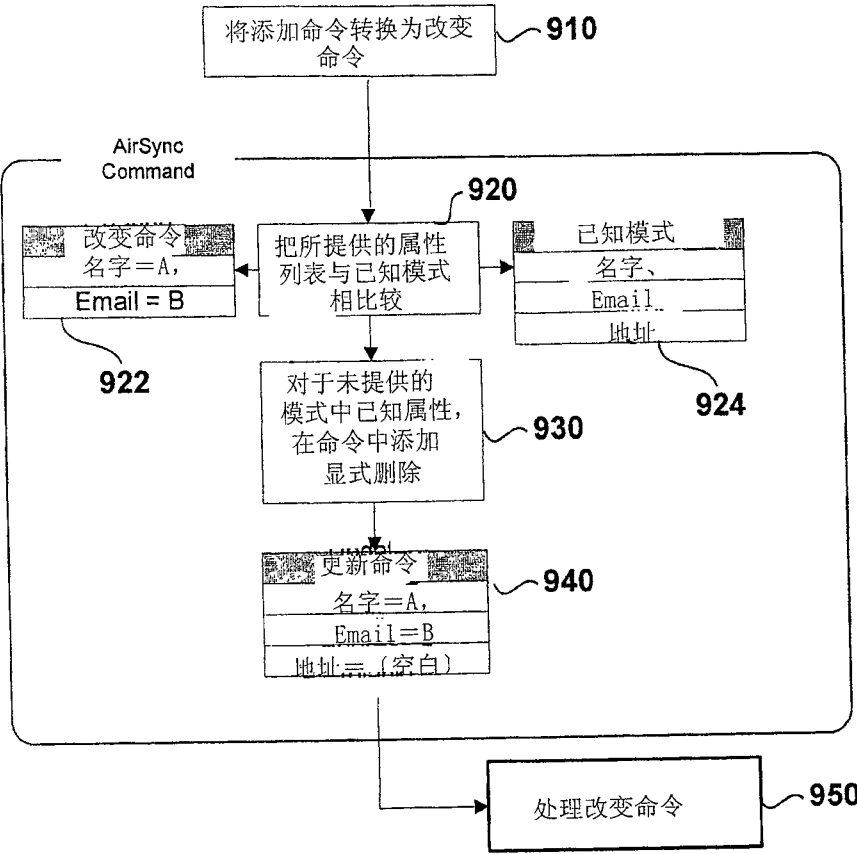


图 9

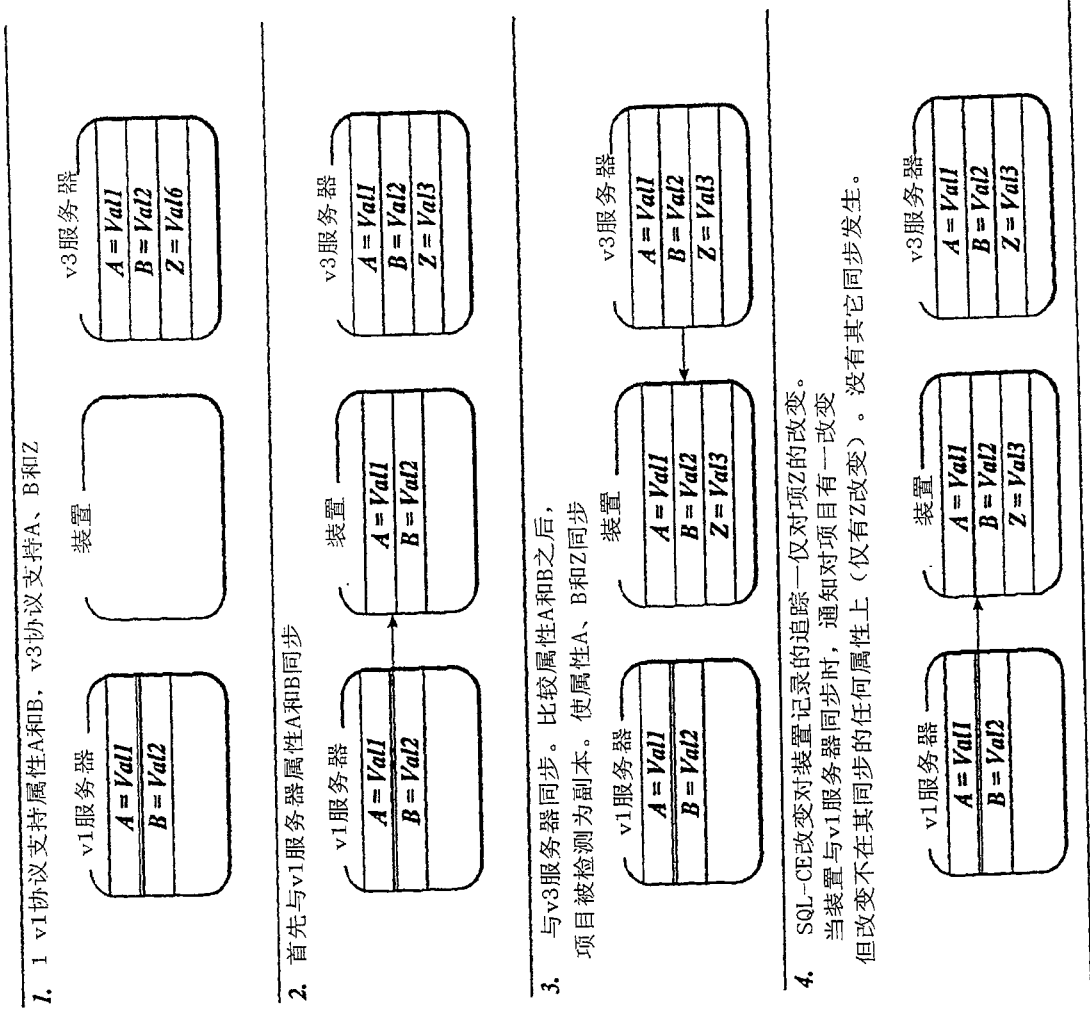
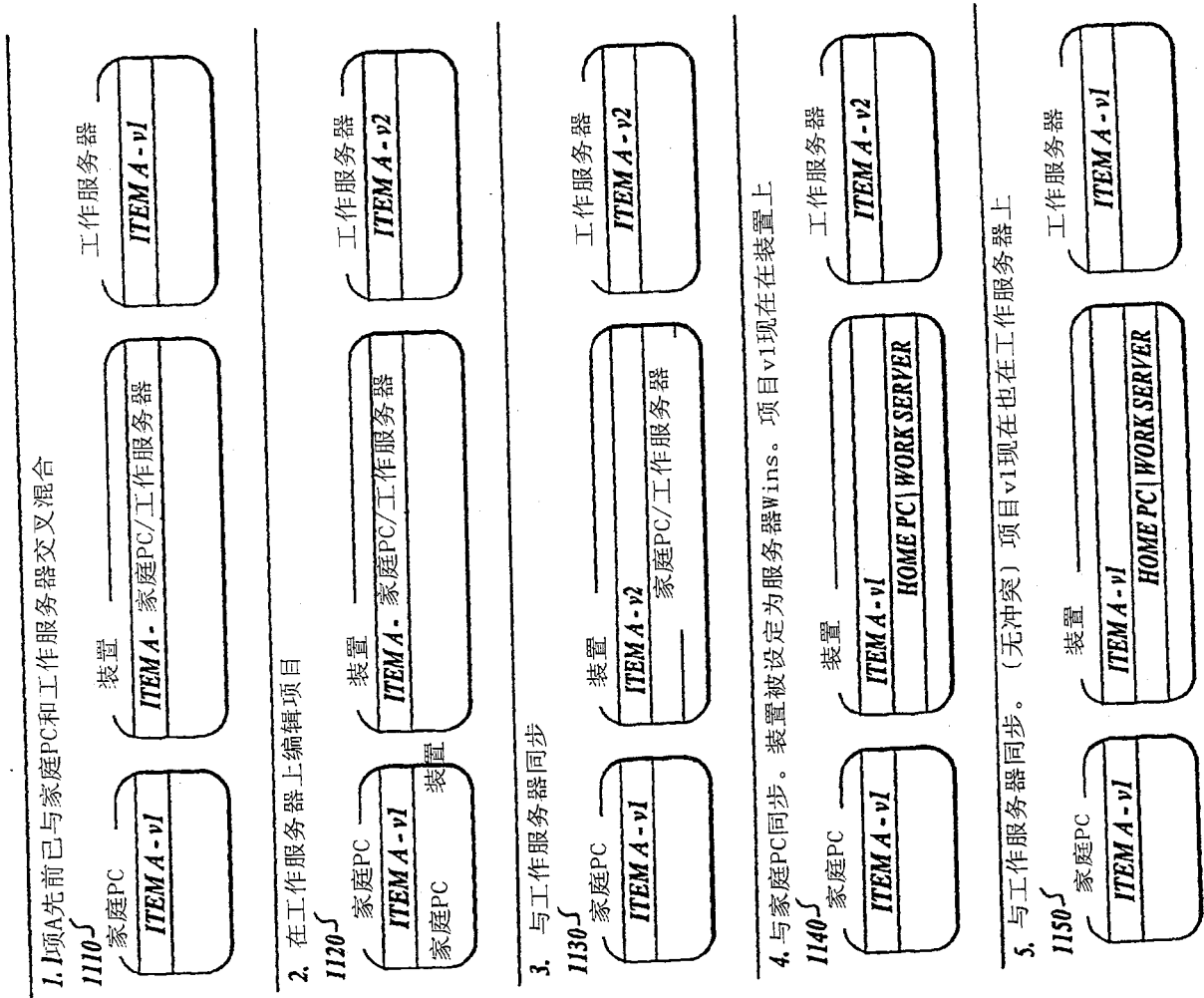


图 10



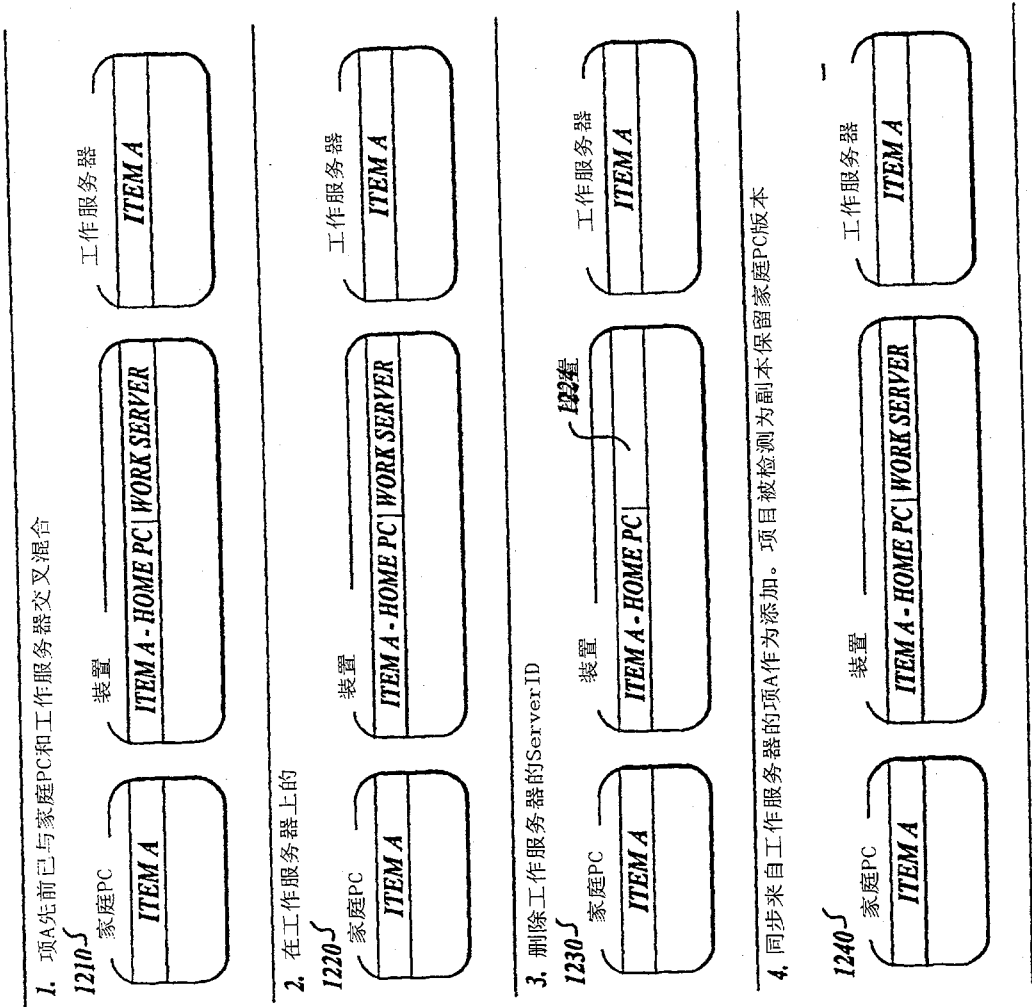


图 12

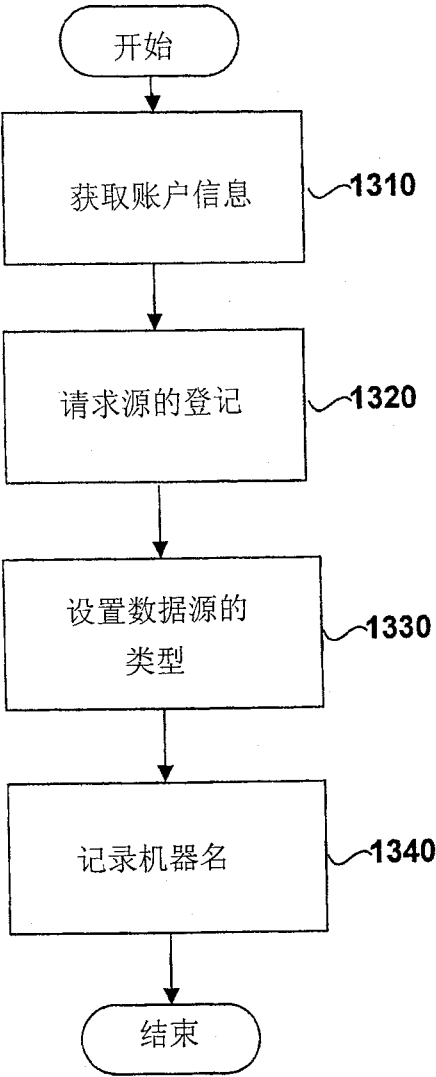


图 13

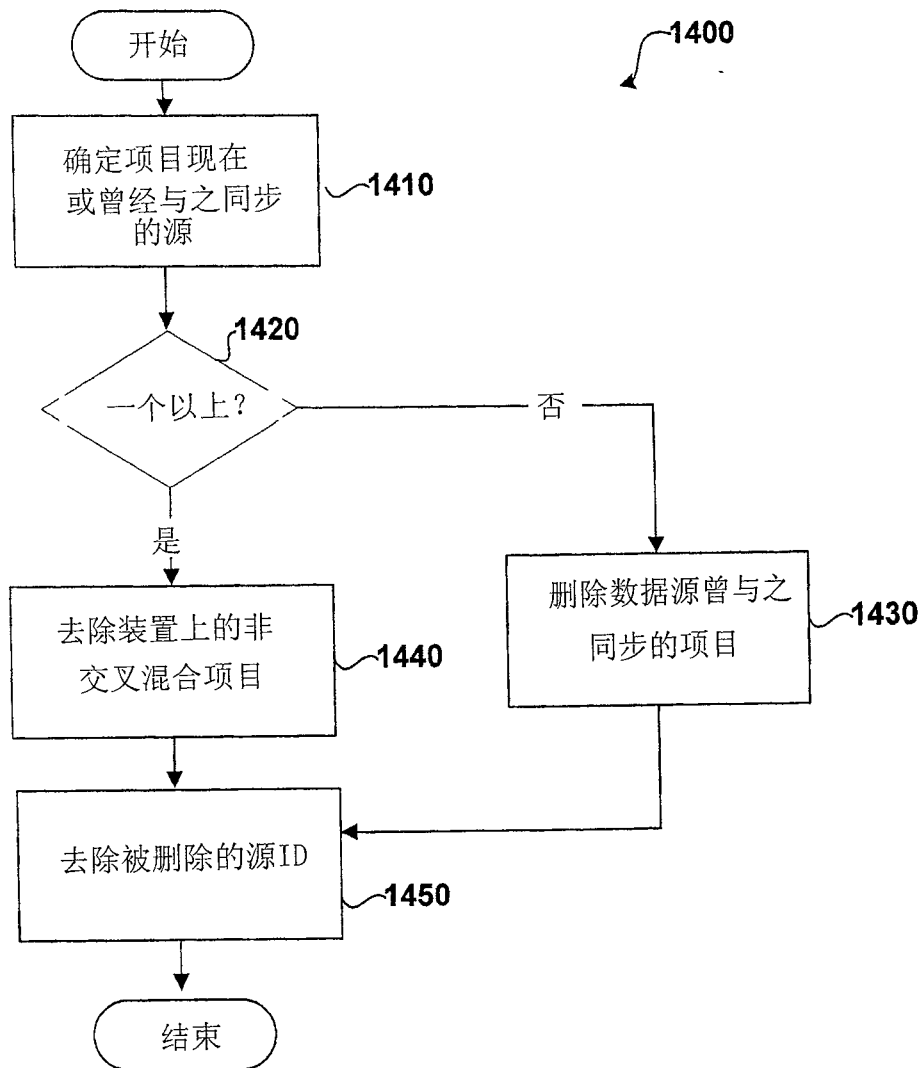


图 14

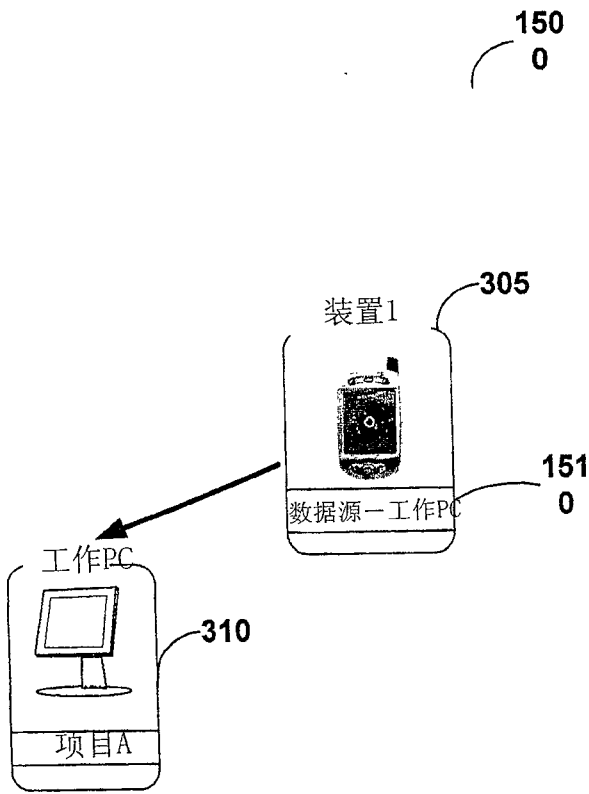


图 15