(19)中华人民共和国国家知识产权局

(12)发明专利

(10)授权公告号 CN 101911010 B
(45)授权公告日 2013.05.08

(21)申请号 200880124599.2
(22)申请日 2008.12.22
(30)优先权数据
11/971,206 2008.01.08 US
(85)PCT申请进入国家阶段日
2010.07.07
(86)PCT申请的申请数据
PCT/US2008/087925 2008.12.22
(87)PCT申请的公布数据
WO2009/088727 EN 2009.07.16
(73)专利权人 微软公司
地址 美国华盛顿州
(72)发明人 A·R·库内奥 B·沃莱恩 E·M·岑茨
(74)专利代理机构 上海专利商标事务所有限公司 31100
代理人 顾嘉运 钱静芳
(51)Int.Cl.
*G06F 9/00*(2006.01)
*G06F 9/44*(2006.01)
*G06F 15/00*(2006.01)
(56)对比文件
CN 1859379 A,2006.11.08, 说明书第4,5,7,8页.
US 2006085426 A1,2006.04.20, 全文.
CN 101052956 A,2007.10.10, 全文.
CN 101040250 A,2007.09.19, 全文.
US 6907428 B2,2005.06.14, 全文.
审查员 解欣

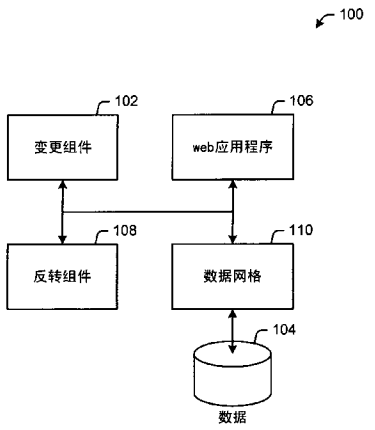
权利要求书2页 说明书8页 附图14页

(54)发明名称

JAVASCRIPT网格中的异步多级撤消支持

(57)摘要

一种用在基于网格的应用程序中的客户机上的多级撤消的体系结构。该体系结构是控制驱动的级联变更系统,其中变更跟踪在异步(以及同步)场景中无缝地工作。客户机应用程序与网格对象相关联,并且实例化并配置该网格对象。该应用程序可以启动对网格中的数据的变更和/或用户可以直接编辑网格中的数据。变更的结果是对应用程序的通知,该通知包括次序键。该应用程序消费该通知并随后可以通过使用该次序键调用更新函数来基于同步或异步计算追加新变更。该应用程序使用该键来附加正确地收集在一起以用撤消/重做的进一步更新。




CN 101911010 B

1. 一种计算机实现的撤消系统，包括：

用于跟踪经由 web 应用程序对数据的异步数据变更的变更组件，其中所述异步数据变更是需要异步确认或扩充的变更；

用于对所述异步数据变更执行多级撤消 / 重做反转操作的反转组件，所述反转组件包括存储显式变更和隐式变更的撤消栈，其中撤消命令从所述撤消栈移除并还原变更直到一显式变更被还原为止；

用于提供存储在所述 web 应用程序中的数据的表格表示的数据网格，用以支持对所述异步数据变更的编辑和可视化，所述数据网格包括用于记录所述异步数据变更的单元格，所述数据网格向所述 web 应用程序发送与所述异步数据变更相关联的通知，所述通知包括次序键，所述次序键便于在所述网格处所述变更相对于其他数据变更的排序。

2. 如权利要求 1 所述的系统，其特征在于，所述数据变更与网页相关联。

3. 如权利要求 1 所述的系统，其特征在于，所述 web 应用程序是浏览器应用程序。

4. 如权利要求 1 所述的系统，其特征在于，所述数据变更是经由基于客户机的数据网格来作出的。

5. 如权利要求 4 所述的系统，其特征在于，所述数据变更是经由所述网格来手动地或通过程序作出的。

6. 如权利要求 1 所述的系统，其特征在于，所述反转操作将所述数据恢复到根据至少两个先前变更而存在的状态。

7. 如权利要求 1 所述的系统，其特征在于，所述 web 应用程序接收所述通知。

8. 如权利要求 1 所述的系统，其特征在于，所述变更与服务器文档相关联，其被输入基于客户机的撤消栈以用于所述反转操作并在基于客户机的变更跟踪器中进行跟踪。

9. 一种提供数据反转操作的计算机实现的方法，包括：

经由基于客户机的网格检测服务器的服务器文档中的数据变更（1000）；

向变更通知中的变更分配次序键（1002）；

将所述变更通知发送到所述服务器以供确认（1004）；

基于所述通知从所述服务器接收异步确认信息（1006）；

根据所述次序键对所述网格中的确认信息进行排序（1008）；以及

基于所述次序键管理所述服务器文档中的撤消 / 重做操作（1010）。

10. 如权利要求 9 所述的方法，其特征在于，还包括基于所述网格中的变更通过撤消栈和变更跟踪器来级联变更。

11. 如权利要求 9 所述的方法，其特征在于，还包括作为撤消 / 重做操作的一部分，将变更存储在撤消栈中并且还原所述变更直至到达第一显式变更。

12. 如权利要求 9 所述的方法，其特征在于，还包括将显式变更存储在变更跟踪器数据结构中。

13. 如权利要求 9 所述的方法，其特征在于，还包括还原下一显式动作之前的所有隐式变更并随后还原该下一显式动作。

14. 一种提供数据反转操作的计算机实现的方法，包括：

经由客户机 web 应用程序启动对服务器的 web 文档的数据变更；

将所述变更作为变更的排序历史存储在客户机撤消栈中，以及对于直接影响数据的显

式变更将变更条目存储在一分开的客户机变更跟踪器中，所述撤消栈存储显式变更和隐式变更；

在所述服务器处确认所述变更；

从所述服务器将确认信息异步地接收到所述 web 应用程序中；以及

基于所述撤消栈中的变更的排序历史和所述变更跟踪器中的变更条目来管理所述 web 文档中的撤消 / 重做操作，其中撤消命令从所述撤消栈移除并还原变更直到一显式变更被还原为止。

15. 如权利要求 14 所述的方法，其特征在于，还包括经由客户机数据网格来展示所述数据变更和所述数据变更的可视化。

16. 如权利要求 14 所述的方法，其特征在于，还包括撤消所述 web 文档中的在所述 web 应用程序中执行的保存操作之前发生的变更。

17. 如权利要求 14 所述的方法，其特征在于，还包括响应于保存操作，从所述变更跟踪器读出变更，将所述变更提交给数据源，并清除所述变更跟踪器。

18. 如权利要求 14 所述的方法，其特征在于，所述变更的排序历史是基于分配给每一变更的变更键的。

19. 如权利要求 14 所述的方法，其特征在于，还包括将隐式变更信息和显式变更信息连同用于执行来撤消隐式和显式变更的函数一起存储在所述撤消栈中。

20. 如权利要求 14 所述的方法，其特征在于，还包括导出解释变更来作为单元格级变更的集合。

# JAVASCRIPT 网格中的异步多级撤消支持

[0001] 背景

[0002] 桌面客户机应用程序和“瘦”web 应用程序之间的主要区别之一是编辑体验的丰富性。传统上，web 应用程序根据各单独的回发 (post-back) 来与服务器办理数据，这些回发在用户导航该应用程序时将数据提交给服务器。一旦用户导航离开网页，则撤消用户动作通常是不可能的。相反，用户可以更流畅地与客户机应用程序交互，只有在用户准备好时才保存数据。此外，如果用户在客户机应用程序中工作时犯了编辑错误，则该用户可以选择“撤消”一次或多次来还原变更，从而不影响所保存的文件。

[0003] 许多软件技术现在需要雇员通过 web 类应用程序来与企业服务器上的数据进行交互。考虑对通常在项目服务器中例如经由网页找到的结构化任务数据的编辑（例如，添加 / 删除任务、分配资源、改变排定的数据等）。在没有多级撤消能力的情况下，这一体验是危险的，因为不允许用户撤消动作。用户可以执行频繁的保存，但将这样的数据集保存到服务器是缓慢的过程。因此，有效的编辑性能与用户对将复杂数据键入网页的舒适度成比例地降低。

[0004] 概述

[0005] 以下提出了简化概述以便提供对在此描述的某些新颖实施例的基本理解。该概述不是详尽的概览，它不旨在标识关键 / 重要的元素，也不旨在描绘其范围。其唯一的目的是以简化的形式来介绍一些概念，作为稍后提出的更为详细的描述的序言。

[0006] 所公开的多级撤消体系结构是控制驱动的级联变更系统，其中变更跟踪在异步（以及同步）场景中无缝地工作。此外，撤消超越保存动作，理解隐式和显式变更之间的差异，并相应地处理这些变更以提供用于撤消的上下文。

[0007] 客户机应用程序与将该应用程序的数据呈现为表格表示的网格对象相关联，从而支持编辑和可视化。该应用程序部分地实例化并配置该网格对象。该应用程序可以通过程序启动对网格中的数据的变更和 / 或用户可以直接编辑网格中的数据。变更的结果是对应用程序的通知，该通知包括次序键。该应用程序消费该通知并随后可以通过使用该次序键调用更新函数来基于同步或异步计算（例如，调度）追加新变更。该应用程序可以在将来的任何时候使用该键来附加对变更的将来更新。这些更新被正确地收集在一起以用于撤消 / 重做。

[0008] 为实现上述及相关目的，本文结合下面的描述和附图描述某些说明性方面。然而，这些方面仅指示了可利用此处公开的原理的各种方法中的少数几种，且旨在包括所有这些方面及等效方面。结合附图阅读下面的详细描述，则其他优点和新颖特征将变得清楚。

[0009] 附图简述

[0010] 图 1 示出计算机实现的撤消系统。

[0011] 图 2 示出提供多级数据变更反转操作的客户机 - 服务器系统的实现。

[0012] 图 3 示出对数据的异步撤消 / 重做操作的示例。

[0013] 图 4 示出基于对变更键的使用来解决异步乱序变更处理的最终数据更新。

[0014] 图 5 示出作为变更组件的一部分的变更跟踪器和作为反转组件的一部分的撤消

栈。

[0015] 图 6 示出网格中的数据的初始网格状态、撤消栈的栈状态、以及变更跟踪器的跟踪器状态。

[0016] 图 7 示出隐式变更和对栈状态及跟踪器状态的影响。

[0017] 图 8 示出网格数据的任务名称的变更和对撤消栈及变更跟踪器的影响。

[0018] 图 9 示出撤消操作的执行。

[0019] 图 10 示出提供数据反转操作的计算机实现的方法。

[0020] 图 11 示出处理显式 / 隐式变更的方法。

[0021] 图 12 示出

[0022] 图 13 示出根据所公开的体系结构的可用于执行多级撤消的计算系统的框图。

[0023] 图 14 示出用于多级撤消处理的示例性客户机 - 服务器计算环境的示意性框图。

[0024] 详细描述

[0025] 所公开的体系结构通过在基于网格的应用程序中的客户机上支持多级撤消而桥接了“瘦”web 应用程序的编辑体验中的主要空白。这一特征对多个在线编辑体验而言也是有用的。例如，多级撤消允许用户对一次编辑更多数据感到更加舒适，并且增加了用户对应用程序的积极性能的感知。此外，应用程序的在线版本也可获益，使得能撤消和重做记录内的多个变更以便用户具有对编辑的完全控制。

[0026] 现在参照附图，全部附图中，相同的附图标记用于表示相同的元素。在以下描述中，为解释起见，描绘了众多具体细节以提供对本发明的全面理解。然而，显然，各新颖实施例可以在没有这些具体细节的情况下实现。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它们。

[0027] 图 1 示出计算机实现的撤消系统 100。系统 100 包括用于跟踪经由 web 应用程序 106 对数据 104 的异步变更的变更组件 102。系统 100 还包括用于针对对于数据 104 的数据变更执行反转操作（例如，撤消、重做）以返回先前状态的反转组件 108。对数据 104 的变更可以经由数据网格 110 作出。对数据 104 的变更可以经由网格 110 手动地或通过程序来作出。

[0028] 应用程序 106 实例化并配置网格 110。网格 110 是存储在应用程序 106 中的数据的表格表示，从而支持编辑和可视化。变更是一单元格的数据的先前和之后版本加上造成该单元格值变更所需的动作。

[0029] 在一个实施例中，web 应用程序 106 是浏览器应用程序，经由该浏览器应用程序对服务器所主存的网页进行变更。用户随后可针对对于网页文档作出的变更行使多级撤消 / 重做反转操作。在另一实现中，web 应用程序 106 是允许用户与本地数据而非网络数据进行交互并且行使诸如多级撤消 / 重做等反转操作的浏览器。

[0030] 图 2 示出提供多级数据变更反转操作的客户机 - 服务器系统 200 的实现。系统 200 示出图 1 的系统 100，即，用于经由 web 应用程序 106 来跟踪对数据 104 的异步变更的变更组件 102，和用于对数据变更执行反转操作（例如，撤消、重做）以回到先前状态的反转组件 108。数据变更是经由数据网格 110 手动地和 / 或通过程序作出的。在此，数据变更被应用于 web 服务器 202 的 web 文档（例如，网页）200。随着将编辑传递到网格 110，变更通知经由应用程序 106 发送到服务器 202 以供异步确认。一旦得到确认，则这些变更经由应用程

序 106 作为对数据 104 的更新来发送回网格 110。

[0031] 图 3 示出对数据的异步撤消 / 重做操作的示例 300。异步变更是需要任何数量的异步确认或扩充的变更。在实体对网格数据作出变更时，这一变更的结果是对应用程序 106 的通知。这一通知内包含有次序（即，变更）键。应用程序 106 消费该通知并随后可以通过使用该次序键调用更新函数来基于一些同步或异步计算（例如，调度）追加新变更。应用程序 106 可以在将来的任何时候自由使用该次序键来附加对这一变更的将来更新。这些更新被正确地收集在一起以用于撤消 / 重做。

[0032] 在此，表示成 A 和 B 的两个变更被输入到需要异步确认的网格 110。在进行编辑时，网格 110 按次序来捕捉变更。每一变更都用次序键加上标签。例如，用次序键 A（也表示变更键 A）对变更 A 加上标签，并且用变更键 B 来表示第二变更 B（比变更 A 的时间晚）。在进行数据变更时，网格 110 检测到这一点并向应用程序 106（例如，浏览器）发送通知，其中应用程序 106 随后将该通知发送到服务器 202 以供确认。服务器 202 处的确认过程可乱序发生，或者一旦完成确认则服务器可乱序发出先前排序的确认。因此，次序键便于在网格 110 处对变更进行排序。

[0033] 在该示例中，第一变更通知请求 302 从网格 110 发送到应用程序 106 以说明数据变更 A，数据变更 A 是在第二行对具有被标记为“持续时间”的字段的列作出的并且该持续时间的新值被设定成 5 天。在后续数据编辑中，网格 110 将第二变更通知请求 304 发送到应用程序 106 以说明数据变更 B，数据变更 B 是在第三行对具有被标记为“持续时间”的字段的列作出的并且该持续时间的新值被设定成 7 天。变更通知请求（302 和 304）按次序（例如，变更 A 在变更 B 之前）发送到应用程序 106。应用程序 106 随后将通知请求（302 和 304）转发到服务器 202，服务器 202 异步地确认变更请求（302 和 304）并将确认返回给应用程序 106。

[0034] 在此，服务器 202 开始对第一变更通知请求 302 的第一确认过程 306。接着，服务器 202 接收并开始对第二变更通知请求 304 的第二确认过程 308。服务器 202 在第一确认过程 306 之前完成第二确认过程 308。因此，第二更新响应 310 从服务器 202 通过应用程序 106 发送到网格 110 以用于更新相关联的数据。第二更新响应 310 包括表示相对于第一数据变更 A 作出该数据变更的次序的次序键 B。第二更新响应 310 还包括该变更是在第三行在被标记为“结束日期”的字段处作出的以及结束日期的新值“5/27”。这对应于第二变更通知请求 304 中的新值“7 天”。

[0035] 服务器 202 随后完成第一确认过程 306 并将第一更新响应 312 通过应用程序 106 发送到网格 110 以供更新相关联的数据。第一更新响应 312 包括表示相对于第二数据变更 B 作出该数据变更的次序的次序键 A。第一更新响应 312 还包括该变更是在第二行在被标记为“结束日期”的字段处作出的以及结束日期的新值“5/25”。这对应于第一变更通知请求 302 中的新值“5 天”。因此，变更从确认乱序返回（变更 B 在变更 A 之前）。

[0036] 图 4 示出基于对变更键的使用来解决异步乱序处理的最终数据更新。示出了对自从最后保存操作以来发生的数据变更的请求的结果 400。在从服务器 202 返回时，这些结果按变更（即，次序）键所指示的逻辑次序（变更 B 在变更 A 之后）而非事件实际发生的次序（变更 A 在变更 B 之后）来放置和存储。因此，与变更 A 相关的变更 402 被存储在一起并被表示为发生在变更 B 之前。类似地，与变更 B 相关的变更 404 被存储在一起并被表示

为发生在变更 A 之后。

[0037] 撤消和重做反转操作也以此方式处理。因此，如果用户要选择撤消，则即使对变更 A 的更新是网格遇到的最后事件，与变更 B 相关联的两个变更（持续时间和结束日期）也将被撤消，因为这是用户作出的最后变更，如变更键 B 所示。

[0038] 实体（例如，用户或系统）与网格交互以对该网格中的数据显式地或隐式地作出变更。显式变更是对网格作出的直接影响数据的变更（例如，变更任务的开始日期）。隐式变更是对网格作出的对数据没有影响的变更（例如，调整列的大小）。

[0039] 图 5 示出作为变更组件 102 的一部分的变更跟踪器 500 和作为反转组件 108 的一部分的撤消栈 502。在实体对网格数据作出变更时，记录每一变更的类型，并且将用于执行和撤消的函数置于撤消栈 502 上。撤消栈 502 是存储最后动作集合的排序历史的数据结构。在撤消事件发生时，从撤消栈 502 移除操作并还原操作，直至遇到第一显式变更为止。该第一显式变更是所还原的最后动作。此时，撤消操作结束。以相同的方式处理后续撤消操作；撤消命令将在下一显式动作之前还原所有隐式动作，并且随后还原显式动作。

[0040] 这一系统的效果是在撤消操作后，网格的视觉状态恢复到被撤消的动作之前。所有显式变更都被存储在与撤消栈 502 分开的结构（变更跟踪器 500）中。这一变更跟踪器 500 可作为单元格级变更的集合来导出。

[0041] 撤消栈 502 和变更跟踪器 500 是独立的结构。在变更发生时，该变更被推送到撤消栈 502，并且如果该变更是显式的则在变更跟踪器 500 中创建对应于该变更的条目。如果该变更是隐式的，则不在变更跟踪器 500 中记录该变更。

[0042] 在撤消事件发生时，从撤消栈 502 和变更跟踪器 500 中移除该变更，并且在变更跟踪器 500 中记录新值。如果该变更存在于变更跟踪器 500 中，则移除该变更。如果该变更不存在于变更跟踪器 500 中，则将用于还原该变更的动作（其本身是变更）添加到变更日志。在保存操作发生时，从变更跟踪器 500 读出变更并将其提交给数据源，并且清除变更跟踪器 500。

[0043] 这一系统的效果是实体可以撤消在保存事件之前发生的动作，因为撤消变更所必需的信息被存储在撤消栈 502 中。类似地，变更跟踪器 500 可以在没有撤消栈 502 的情况下工作。

[0044] 图 6-9 示出用于例示在网格数据状态、撤消栈、以及变更跟踪器中发生的变更的一系列示图。图 6 示出网格中的数据的初始网格状态 600、撤消栈的栈状态 602、以及变更跟踪器的跟踪器状态 604。变更将在网格数据的第二和第三行中发生。

[0045] 实体随后通过决定为围栏上漆应花 2 天而非 1 天来作出显式变更。变更网格状态 606 包括第二行持续时间列中的变为 2 天的变更。在作为项目管理应用程序的一部分的项目的情况下，例如，持续时间的变更造成结束日期（也被称为“结束”）的变更（级联变更）。创建涵盖这两个变更的级联事务。因此，第二行中“结束”列中的字段信息从“星期一 5/21/xx”变更为“星期二 5/22/xx”，如变更的网格状态 606 所示。

[0046] 这些持续时间和结束变更被推送到撤消栈，如栈状态 608 所示。另外，跟踪器状态 610 反映这些变更。图 7 示出隐式变更和对栈状态及跟踪器状态的影响。在此，实体隐藏“结束”列，从而变更到网格状态 612，这将该变更推送到撤消栈，从而变更到栈状态 614，但不变更变更跟踪器的跟踪器状态 610。实体保存该项目，这使得变更跟踪器中的跟踪器状态

610 被清除成栈状态 616，因为网格的数据现在与服务器数据相一致。

[0047] 图 8 示出网格数据的任务名称的变更和对撤消栈及变更跟踪器的影响。实体基于第三行中任务名称从“清理工具”到“清理项目”的变更来变更网格状态 618。栈状态 620 反映被推送到栈的这一变更，并且跟踪器反映跟踪器状态 622。

[0048] 图 9 示出撤消操作的执行。撤消通过从撤消栈和变更跟踪器弹出变更来还原最后的显式变更，如在栈状态 622 和跟踪器状态 626 所示（之前没有变更）。实体随后再一次执行撤消，这还原隐式变更和其余的显式变更。这一级联效果将变更从撤消栈移除到栈状态 628，将对变更跟踪器的反转变更添加到跟踪器状态 630，并将网格状态变更回网格状态 632，其与图 6 网格状态 600 相同。

[0049] 以下是表示用于执行所公开的体系结构的各新颖方面的示例性方法的一系列流程图。尽管出于解释简明的目的，此处例如以流图或流程图形式示出的一个或多个方法被示出并描述为一系列动作，但是可以理解和明白，各方法不受动作的次序的限制，因为根据本发明，某些动作可以按与此处所示并描述的不同的次序和 / 或与其他动作同时发生。例如，本领域技术人员将会明白并理解，方法可被替换地表示为一系列相互关联的状态或事件，诸如以状态图的形式。此外，并非在一方法中示出的所有动作都是新颖实现所必需的。

[0050] 图 10 示出提供数据反转操作的计算机实现的方法。在 1000，经由基于客户机的网格检测服务器的服务器文档中的数据变更。在 1002，向变更通知中的变更分配次序键。在 1004，将变更通知发送到服务器以供确认。在 1006，基于通知从服务器接收异步确认信息。在 1008，在网格中根据次序键对确认信息进行排序。在 1010，基于次序键管理服务器文档中的撤消 / 重做操作。

[0051] 图 11 示出处理显式 / 隐式变更的方法。在 1100，经由网格接收数据变更。在 1102，将变更的类型和用于撤消 / 重做该变更的函数存储在撤消栈中。在 1104，接收反转操作（例如，重做、撤消）。在 1106，从栈中移除撤消操作并还原，直至遇到第一显式变更（其是被还原的最后动作）为止。

[0052] 图 12 示出提供数据反转操作的计算机实现的方法。在 1200，经由客户机 web 应用程序启动对服务器的 web 文档的数据变更。在 1202，将变更作为变更的排序历史存储在客户机撤消栈中并作为变更条目存储在客户机变更跟踪器中。在 1204，在服务器处确认变更。在 1206，从服务器将确认信息异步地接收到 web 应用程序中。在 1208，在客户机处基于撤消栈中的变更的排序历史和变更跟踪器中的变更条目来管理 web 文档中的撤消 / 重做操作。

[0053] 如在本申请中所使用的，术语“组件”和“系统”旨在表示计算机相关的实体，其可以是硬件、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，在处理器上运行的进程、处理器、硬盘驱动器、多个（光和 / 或磁存储介质的）存储驱动器、对象、可执行代码、执行的线程、程序、和 / 或计算机。作为说明，运行在服务器上的应用程序和服务器都可以是组件。一个或多个组件可以驻留在进程和 / 或执行的线程内，且组件可以位于一台计算机上和 / 或分布在两台或更多的计算机之间。

[0054] 现在参考图 13，示出了可用于执行根据所公开的体系结构的多级撤消的计算系统 1300 的框图。为了提供用于其各方面的附加上下文，图 13 及以下讨论旨在提供对其中可实现该各方面的合适的计算系统 1300 的简要概括描述。尽管以上描述是在可在一个或多个

计算机上运行的计算机可执行指令的一般上下文中进行的，但是本领域的技术人员将认识到，新颖实施例也可结合其他程序模块和 / 或作为硬件和软件的组合来实现。

[0055] 一般而言，程序模块包括执行特定任务或实现特定抽象数据类型的例程、程序、组件、数据结构等等。此外，本领域的技术人员可以理解，本发明的方法可用其他计算机系统配置来实施，包括单处理器或多处理器计算机系统、小型计算机、大型计算机、以及个人计算机、手持式计算设备、基于微处理器的或可编程消费电子产品等，其每一个都可操作上耦合到一个或多个相关联的设备。

[0056] 所示各方面也可以在其中某些任务由通过通信网络链接的远程处理设备来执行的分布式计算环境中实施。在分布式计算环境中，程序模块可以位于本地和远程存储器存储设备中。

[0057] 计算机通常包括各种计算机可读介质。计算机可读介质可以是可由计算机访问的任何可用介质，且包括易失性和非易失性介质、可移动和不可移动介质。作为示例而非限制，计算机可读介质可以包括计算机存储介质和通信介质。计算机存储介质包括以存储如计算机可读指令、数据结构、程序模块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。计算机存储介质包括但不限于 RAM、ROM、EEPROM、闪存或者其他存储器技术、CD-ROM、数字视频盘 (DVD) 或其他光盘存储、磁带盒、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并且可以由计算机访问的任何其他介质。

[0058] 再次参考图 13，用于实现各方面的示例性计算系统 1300 包括具有处理单元 1304、系统存储器 1306 和系统总线 1308 的计算机 1302。系统总线 1308 向包括但不限于系统存储器 1306 的各系统组件提供到处理单元 1304 的接口。处理单元 1304 可以是市场上可购买到的各种处理器中的任意一种。双微处理器和其他多处理器体系结构也可用作处理单元 1304。

[0059] 系统总线 1308 可以是若干种总线结构中的任一种，这些总线结构还可互连到存储器总线 ( 带有或没有存储器控制器 )、外围总线、以及使用各类市场上可购买到的总线体系结构中的任一种的局部总线。系统存储器 1306 可包括非易失性存储器 (NON-VOL)1310 和 / 或易失性存储器 1312( 例如随机存取存储器 (RAM))。基本输入 / 输出系统 (BIOS) 可被存储在非易失性存储器 1310( 例如 ROM、EPROM、EEPROM 等 ) 中，其中 BIOS 存储帮助诸如在启动期间在计算机 1302 内的元件之间传输信息的基本例程。易失性存储器 1312 还可包括诸如静态 RAM 等高速 RAM 来用于高速缓存数据。

[0060] 计算机 1302 还包括内置硬盘驱动器 (HDD)1314( 例如，EIDE、SATA)，该内置 HDD 1314 还可被配置成在合适的机壳中外部使用 ；磁软盘驱动器 (FDD)1316( 例如，从可移动磁盘 1318 中读取或向其写入 ) ；以及光盘驱动器 1320( 例如，从 CD-ROM 盘 1322 中读取，或从诸如 DVD 等其他高容量光学介质中读取或向其写入 )。HDD 1314、FDD 1316、以及光盘驱动器 1320 可分别由 HDD 接口 1324、FDD 接口 1326 和光盘驱动器接口 1328 来连接到系统总线 1308。用于外置驱动器实现的 HDD 接口 1324 可包括通用串行总线 (USB) 和 IEEE 1394 接口技术中的至少一种或两者。

[0061] 驱动器及相关联的计算机可读介质提供了对数据、数据结构、计算机可执行指令等的非易失性存储。对于计算机 1302，驱动器和介质容纳适当的数字格式的任何数据的存储。尽管以上对计算机可读介质的描述涉及 HDD、可移动磁盘 ( 例如 FDD) 以及诸如 CD 或

DVD 等可移动光学介质，但是本领域的技术人员应当理解，示例性操作环境中也可使用可由计算机读取的任何其他类型的介质，诸如 zip 驱动器、磁带盒、闪存卡、盒式磁带等等，并且任何这样的介质可包含用于执行所公开的体系结构的新颖方法的计算机可执行指令。

[0062] 多个程序模块可存储在驱动器和易失性存储器 1312 中，包括操作系统 1330、一个或多个应用程序 1332、其他程序模块 1334 和程序数据 1336。该一个或多个应用程序 1332、其他程序模块 1334、以及程序数据 1336 可包括例如变更组件 102、数据 104、web 应用程序 106、反转组件 108、数据网格 110、变更通知 (302 和 304)、更新响应 (310 和 312)、变更跟踪器 500、撤消栈 502、网格状态 (600、606、612、618、以及 632)、栈状态 (602、608、614、620、624、以及 628)、以及跟踪器状态 (604、610、616、622、626、以及 630)。

[0063] 操作系统、应用程序、模块和 / 或数据的全部或部分也可被高速缓存在易失性存储器 1312 中。应该明白，所公开的体系结构可以用市场上可购得的各种操作系统或操作系统的组合来实施。

[0064] 用户可以通过一个或多个有线 / 无线输入设备，例如键盘 1338 和诸如鼠标 1340 等定点设备将命令和信息输入到计算机 1302 中。其他输入设备（未示出）可包括话筒、IR 遥控器、操纵杆、游戏手柄、指示笔、触摸屏等等。这些和其他输入设备通常通过耦合到系统总线 1308 的输入设备接口 1342 连接到处理单元 1304，但也可通过其他接口连接，如并行端口、IEEE1394 串行端口、游戏端口、USB 端口、IR 接口等等。

[0065] 监视器 1344 或其他类型的显示设备也经由诸如视频适配器 1346 等接口来连接到系统总线 1308。除了监视器 1344 之外，计算机通常包括诸如扬声器、打印机等其他外围输出设备（未示出）。

[0066] 计算机 1302 可使用经由有线和 / 或无线通信至一个或多个远程计算机，诸如远程计算机 1348 的逻辑连接在网络化环境中操作。远程计算机 1348 可以是工作站、服务器计算机、路由器、个人计算机、便携式计算机、基于微处理器的娱乐设备、对等设备或其他常见的网络节点，并且通常包括相对于计算机 1302 描述的许多或所有元件，尽管为简明起见仅示出了存储器 / 存储设备 1350。所描绘的逻辑连接包括到局域网 (LAN)1352 和 / 或例如广域网 (WAN)1354 等更大的网络的有线 / 无线连接。这一 LAN 和 WAN 连网环境常见于办公室和公司，并且方便了诸如内联网等企业范围计算机网络，所有这些都可连接到例如因特网等全球通信网络。

[0067] 当在 LAN 连网环境中使用时，计算机 1302 通过有线和 / 或无线通信网络接口或适配器 1356 连接到 LAN 1352。适配器 1356 可以方便到 LAN 1352 的有线和 / 或无线通信，并且还可包括其上设置的用于使用适配器 1356 的无线功能进行通信的无线接入点。

[0068] 当在 WAN 连网环境中使用时，计算机 1302 可包括调制解调器 1358，或连接到 WAN 1354 上的通信服务器，或具有用于通过 WAN 1354，诸如通过因特网建立通信的其他装置。或为内置或为外置以及有线和 / 或无线设备的调制解调器 1358 经由输入设备接口 1342 连接到系统总线 1308。在联网环境中，相对于计算机 1302 所描述的程序模块或其部分可以存储在远程存储器 / 存储设备 1350 中。应该理解，所示网络连接是示例性的，并且可以使用在计算机之间建立通信链路的其他手段。

[0069] 计算机 1302 可用于使用 IEEE 802 标准家族来与有线和无线设备或实体通信，例如在操作上安置成与例如打印机、扫描仪、台式和 / 或便携式计算机、个人数字助理 (PDA)、

通信卫星、任何一件与无线可检测标签相关联的设备或位置（例如，电话亭、报亭、休息室）以及电话进行无线通信（例如，IEEE 802.11 空中调制技术）的无线设备。这至少包括 Wi-Fi（即无线保真）、WiMax 和蓝牙 TM 无线技术。由此，通信可以如对于常规网络那样是预定义结构，或者仅仅是至少两个设备之间的自组织 (ad hoc) 通信。Wi-Fi 网络使用称为 IEEE 802.11x(a、b、g 等等）的无线电技术来提供安全、可靠、快速的无线连接。Wi-Fi 网络可用于将计算机彼此连接、连接到因特网以及连接到有线网络（使用 IEEE 802.3 相关介质和功能）。

[0070] 现在参考图 14，示出了用于多级撤消的示例性客户机 - 服务器计算环境 1400 的示意性框图。环境 1400 包括一个或多个客户机 1402。客户机 1402 可以是硬件和 / 或软件（例如，线程、进程、计算设备）。例如，客户机 1402 可容纳 cookie 和 / 或相关联的上下文信息。

[0071] 环境 1400 还包括一个或多个服务器 1404。服务器 1404 也可以是硬件和 / 或软件（例如，线程、进程、计算设备）。服务器 1404 可以例如通过使用本体系结构来容纳线程以执行变换。在客户机 1402 和服务器 1404 之间的一种可能的通信能够以适合在两个或多个计算机进程之间传输的数据分组的形式进行。数据分组可包括例如 cookie 和 / 或相关联的上下文信息。环境 1400 包括可以用来使客户机 1402 和服务器 1404 之间通信更容易的通信框架 1406（例如，诸如因特网等全球通信网络）。

[0072] 通信可经由有线（包括光纤）和 / 或无线技术来促进。客户机 1402 操作上被连接到可以用来存储对客户机 1402 本地的信息（例如，cookie 和 / 或相关联的上下文信息）的一个或多个客户机数据存储 1408。同样地，服务器 1404 可在操作上连接到可以用来存储对服务器 1404 本地的信息的一个或多个服务器数据存储 1410。

[0073] 客户机 1402 可包括 web 应用程序 106，并且客户机数据存储 1408 可包括数据 104。服务器 1404 可包括服务器 202 和异步确认过程 (306 和 308)，并且服务器数据存储 1410 可包括文档 200。

[0074] 上面描述的包括所公开的体系结构的各示例。当然，描述每一个可以想到的组件和 / 或方法的组合是不可能的，但本领域内的普通技术人员应该认识到，许多其他组合和排列都是可能的。因此，该新颖体系结构旨在涵盖所有这些落入所附权利要求书的精神和范围内的更改、修改和变化。此外，就在说明书或权利要求书中使用术语“包括”而言，这一术语旨在以与术语“包含”在被用作权利要求书中的过渡此时所解释的相似的方式为包含性的。

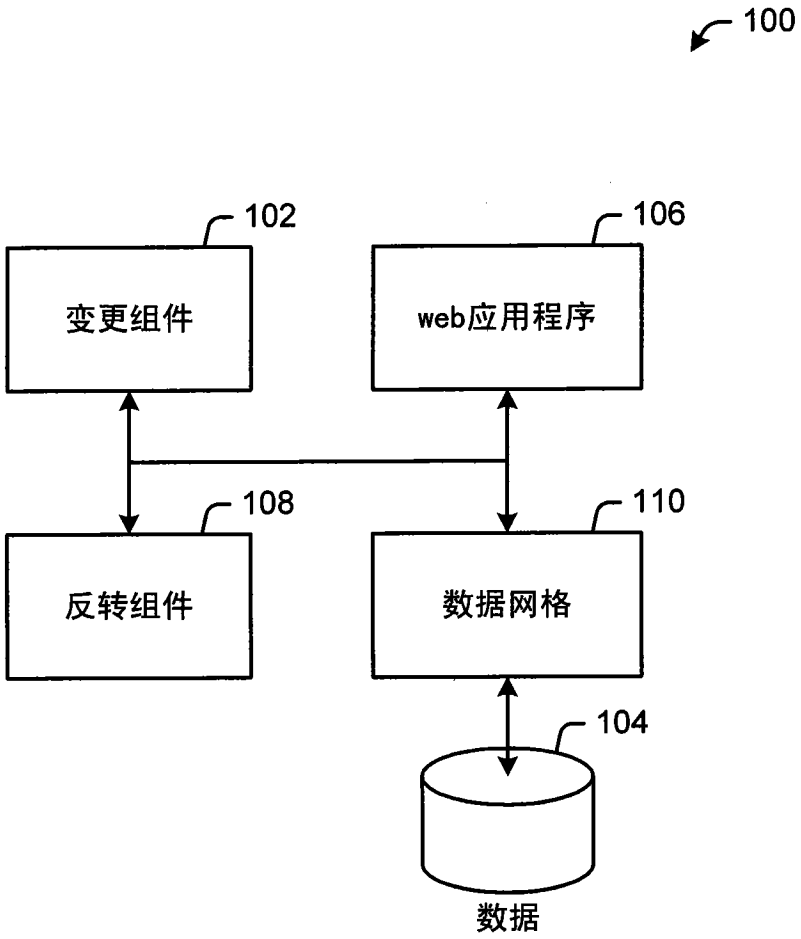



图 1

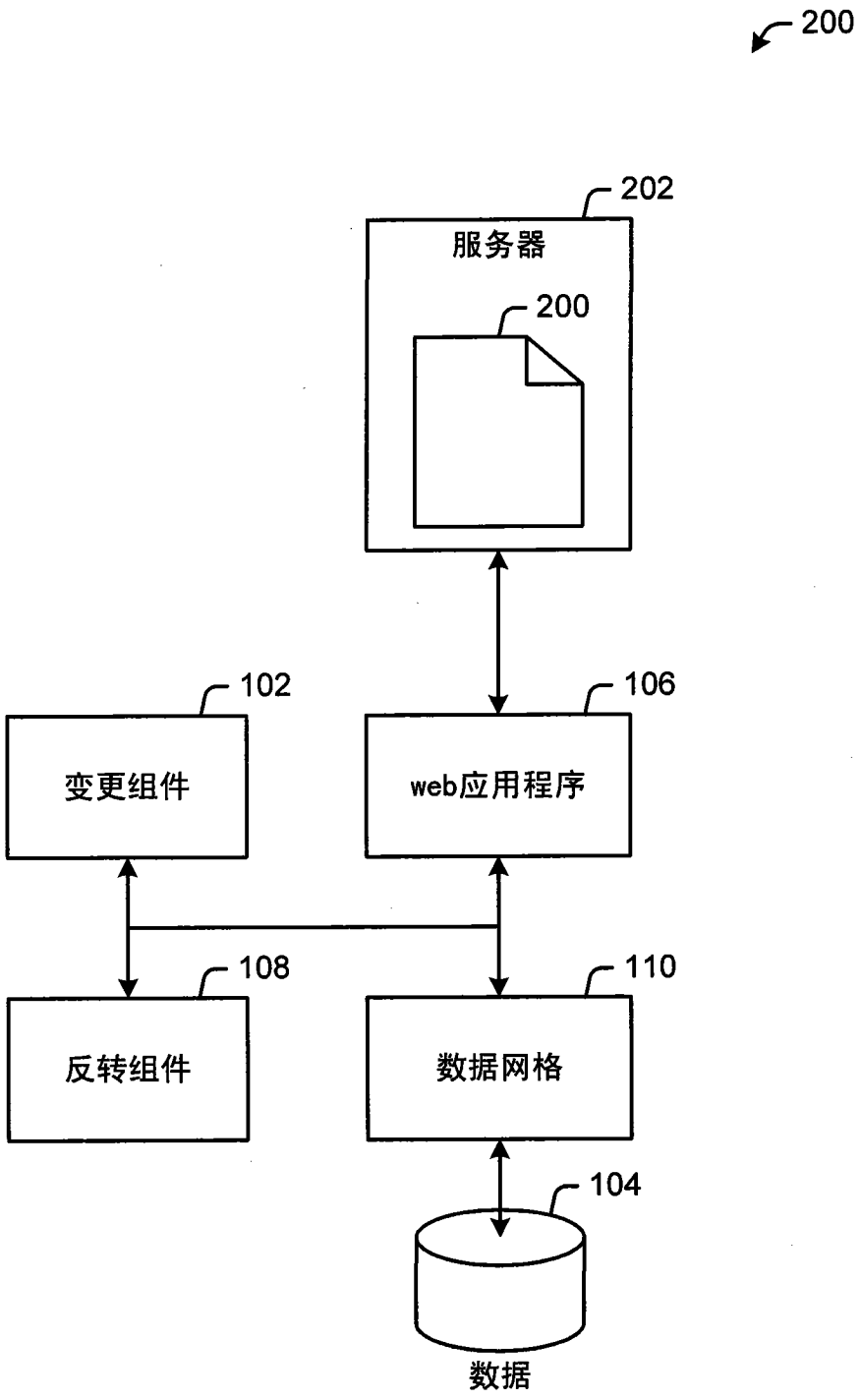



图 2

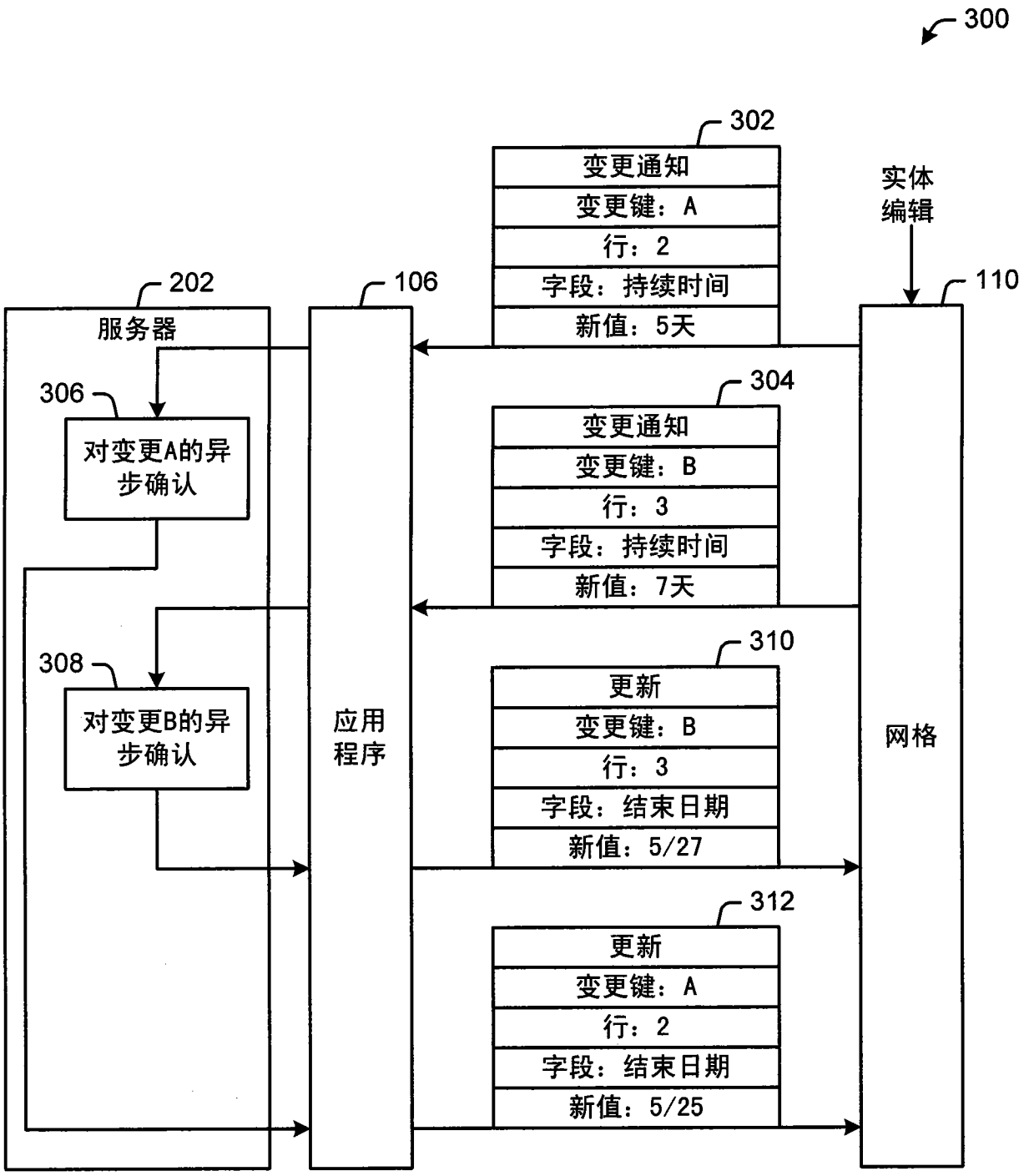

300
302
变更通知
变更键：A
行：2
字段：持续时间
新值：5天
实体
编辑
110
202
服务器
106
306
对变更A的异步确认
304
变更通知
变更键：B
行：3
字段：持续时间
新值：7天
308
对变更B的异步确认
应用程序
310
更新
变更键：B
行：3
字段：结束日期
新值：5/27
网格
312
更新
变更键：A
行：2
字段：结束日期
新值：5/25

图 3

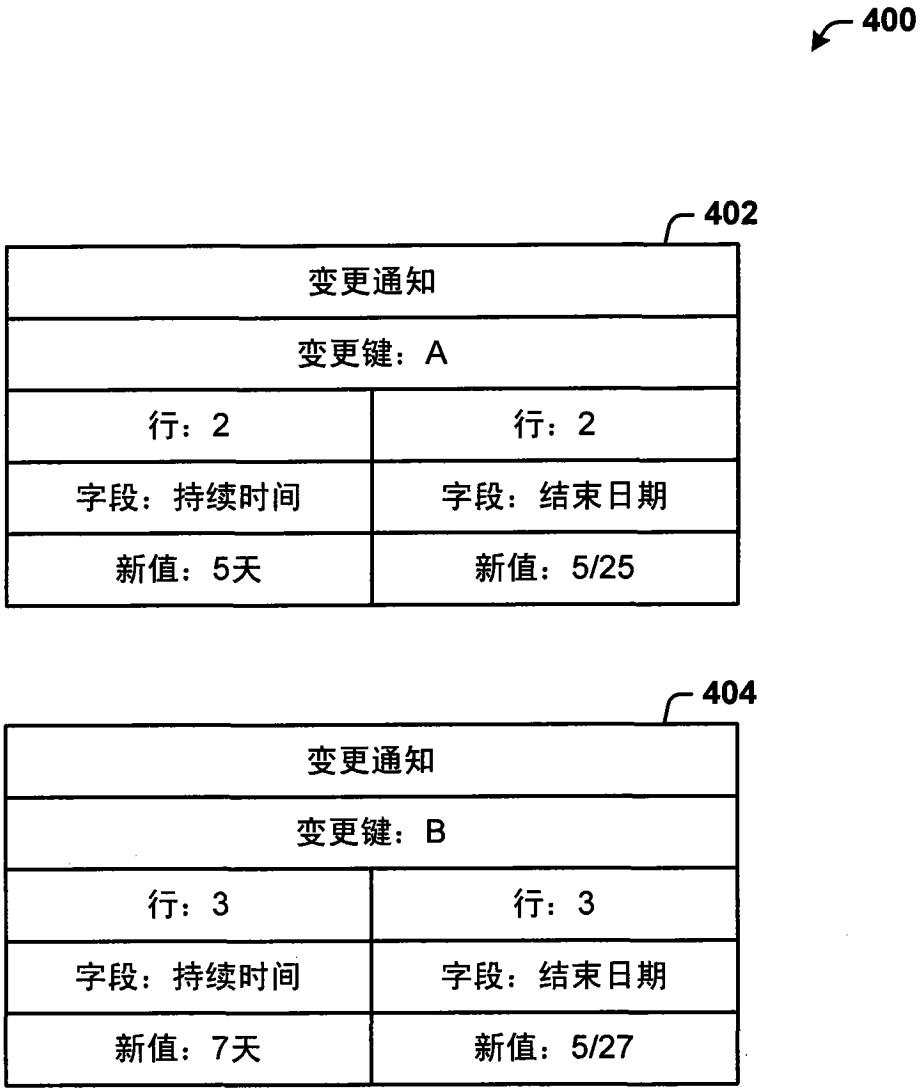

400
402
变更通知
变更键：A
行：2
行：2
字段：持续时间
字段：结束日期
新值：5天
新值：5/25
404
变更通知
变更键：B
行：3
行：3
字段：持续时间
字段：结束日期
新值：7天
新值：5/27

图 4

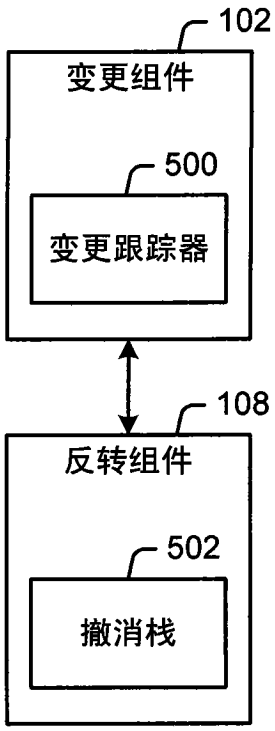



图 5

600
网格状态
任务名称 起始 结束 持续时间
1 发送围栏 星期一5/21/XX 星期一5/21/XX 1 天
2 为围栏上漆 星期一5/21/XX 星期一5/21/XX 1 天
3 清理工具 星期一5/21/XX 星期一5/21/XX 1 天
602
撤消栈的状态
604
变更跟踪器的状态
606
网格状态
任务名称 起始 结束 持续时间
1 发送围栏 星期一5/21/XX 星期一5/21/XX 1 天
2 为围栏上漆 星期一5/21/XX 星期二 5/22/XX 2 天
3 清理工具 星期一5/21/XX 星期一5/21/XX 1 天
608
撤消栈的状态
执行:
更新单元格{{2,持续时间},2}
撤消:
更新单元格{{2,持续时间},1}
执行:
更新单元格{{2,结束},5/23}
撤消:
更新单元格{{2,结束},5/22}
610
变更跟踪器的状态
{2,持续时间}->2
{2,结束}->5/23

图 6

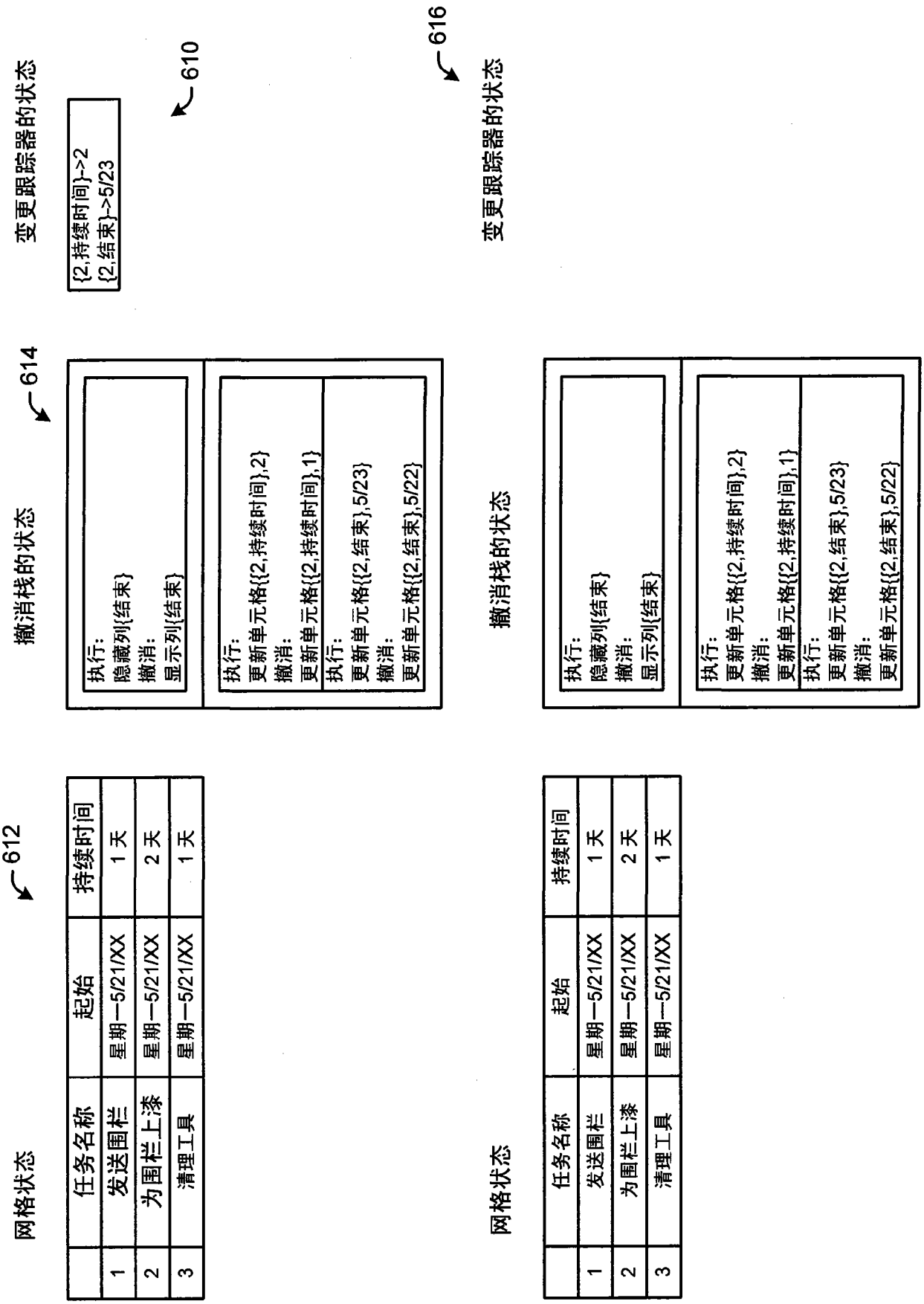
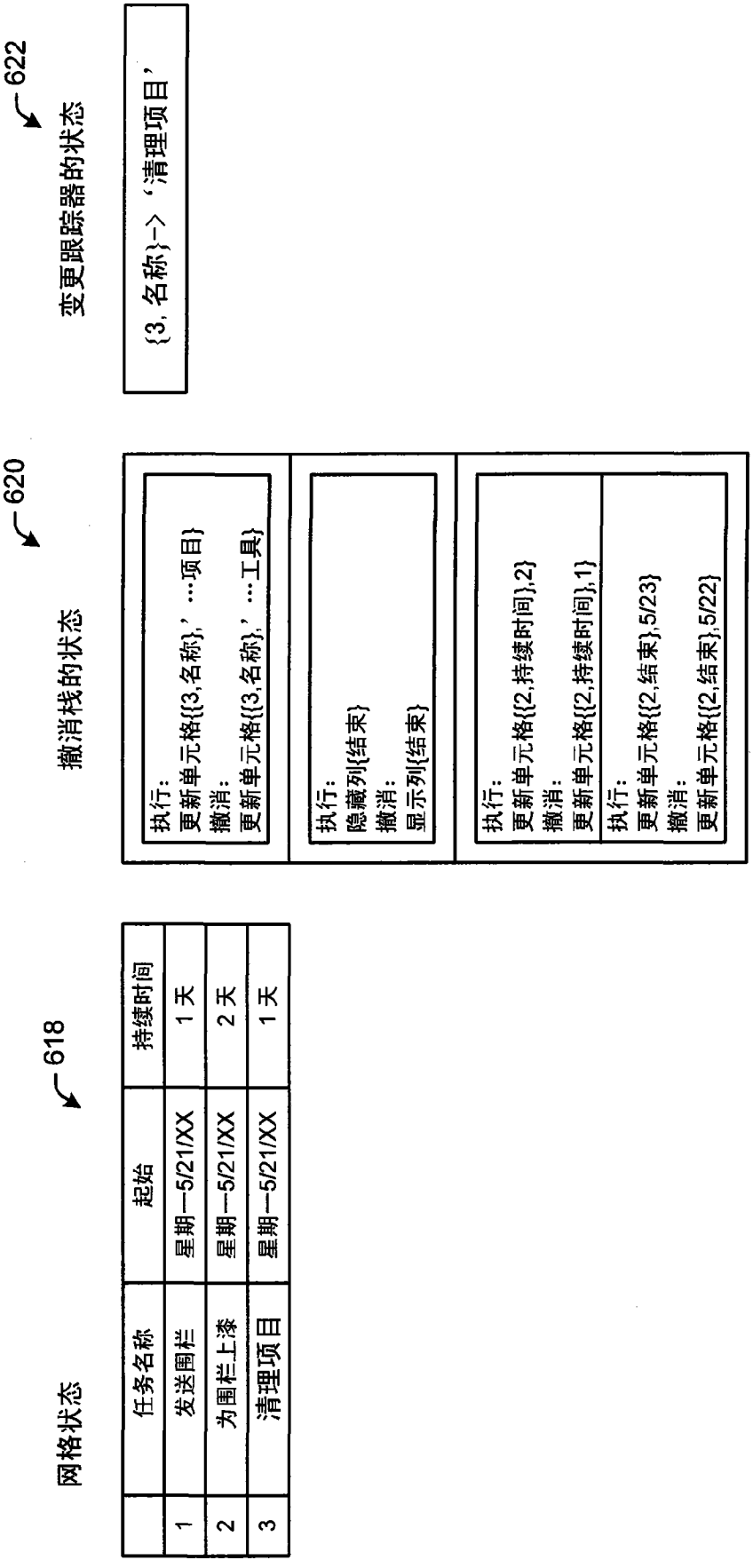

网格状态
612
任务名称
起始
持续时间
1
发送围栏
星期一5/21/XX
1 天
2
为围栏上漆
星期一5/21/XX
2 天
3
清理工具
星期一5/21/XX
1 天
撤消栈的状态
614
执行:
隐藏列{结束}
撤消:
显示列{结束}
执行:
更新单元格{{2,持续时间},2}
撤消:
更新单元格{{2,持续时间},1}
执行:
更新单元格{{2,结束},5/23}
撤消:
更新单元格{{2,结束},5/22}
变更跟踪器的状态
610
{2,持续时间}->2
{2,结束}->5/23
网格状态
任务名称
起始
持续时间
1
发送围栏
星期一5/21/XX
1 天
2
为围栏上漆
星期一5/21/XX
2 天
3
清理工具
星期一5/21/XX
1 天
撤消栈的状态
执行:
隐藏列{结束}
撤消:
显示列{结束}
执行:
更新单元格{{2,持续时间},2}
撤消:
更新单元格{{2,持续时间},1}
执行:
更新单元格{{2,结束},5/23}
撤消:
更新单元格{{2,结束},5/22}
变更跟踪器的状态
616

图 7




| | 任务名称 | 起始 | 持续时间 |
|---|---|---|---|
| 1 | 发送围栏 | 星期一5/21/XX | 1 天 |
| 2 | 为围栏上漆 | 星期一5/21/XX | 2 天 |
| 3 | 清理项目 | 星期一5/21/XX | 1 天 |

图 8

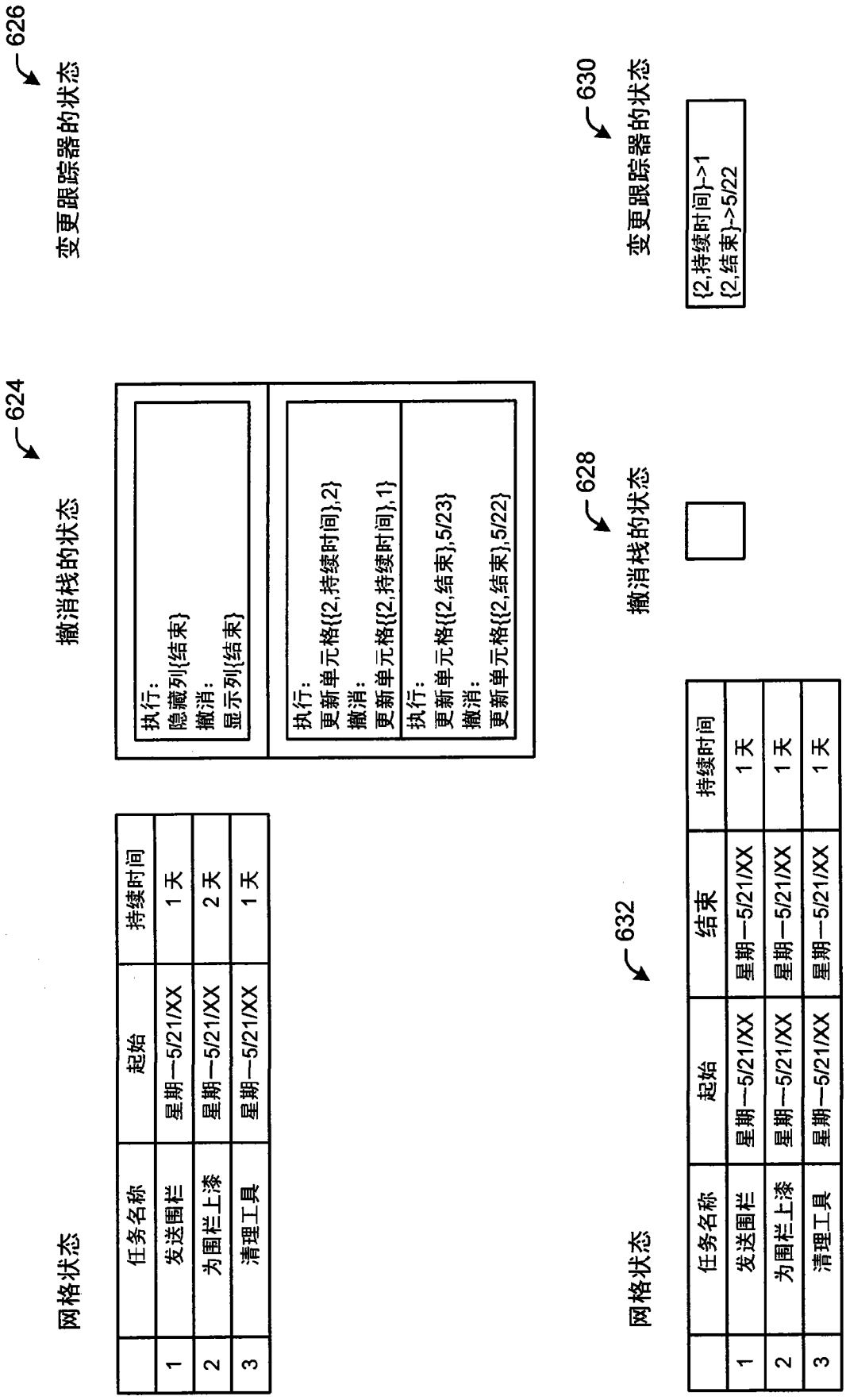

626
变更跟踪器的状态
624
撤消栈的状态
执行:
隐藏列{结束}
撤消:
显示列{结束}
执行:
更新单元格{{2,持续时间},2}
撤消:
更新单元格{{2,持续时间},1}
执行:
更新单元格{{2,结束},5/23}
撤消:
更新单元格{{2,结束},5/22}
网格状态
任务名称 起始 持续时间
1 发送围栏 星期一5/21/XX 1 天
2 为围栏上漆 星期一5/21/XX 2 天
3 清理工具 星期一5/21/XX 1 天
630
变更跟踪器的状态
{2,持续时间}->1
{2,结束}->5/22
628
撤消栈的状态
632
网格状态
任务名称 起始 结束 持续时间
1 发送围栏 星期一5/21/XX 星期一5/21/XX 1 天
2 为围栏上漆 星期一5/21/XX 星期一5/21/XX 1 天
3 清理工具 星期一5/21/XX 星期一5/21/XX 1 天

图 9

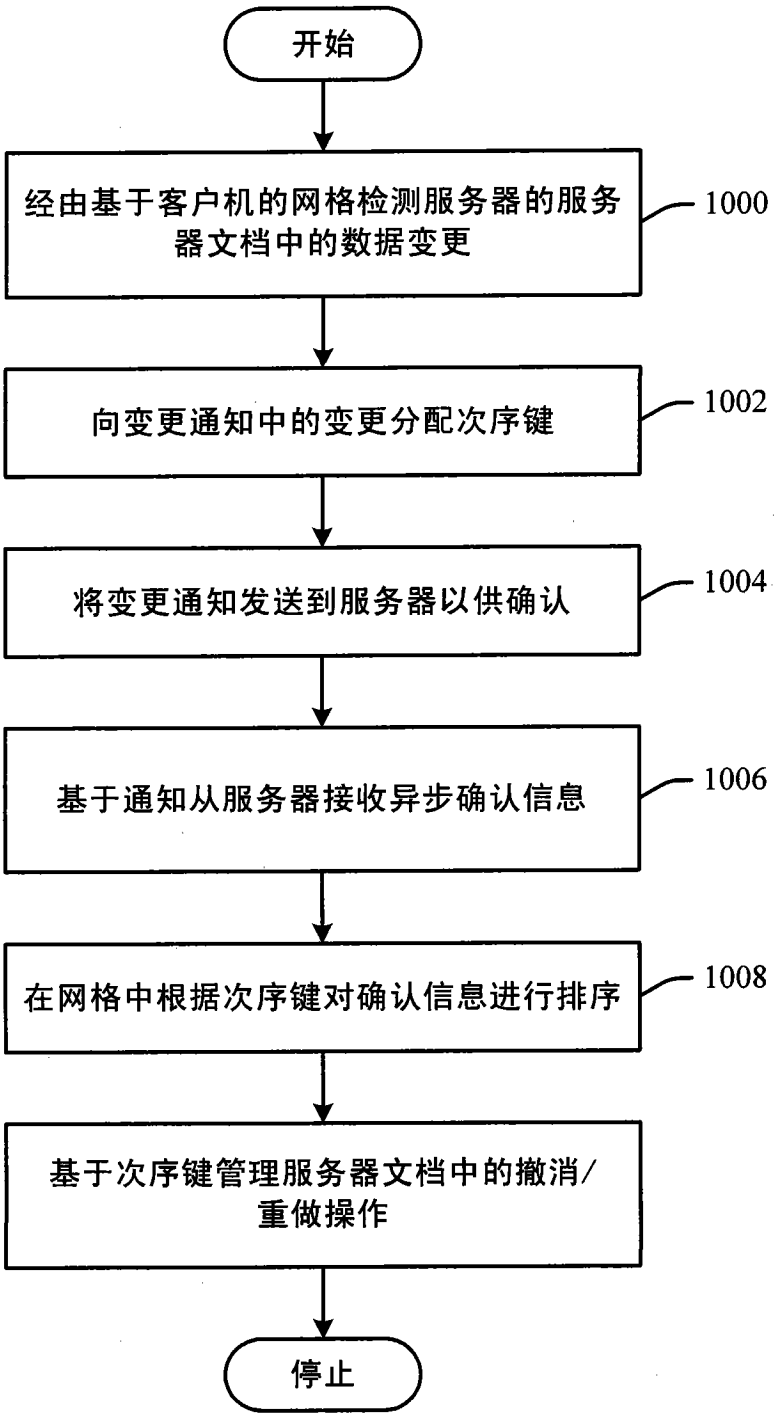

开始
经由基于客户机的网格检测服务器的服务器文档中的数据变更
1000
向变更通知中的变更分配次序键
1002
将变更通知发送到服务器以供确认
1004
基于通知从服务器接收异步确认信息
1006
在网格中根据次序键对确认信息进行排序
1008
基于次序键管理服务器文档中的撤消/重做操作
停止

图 10

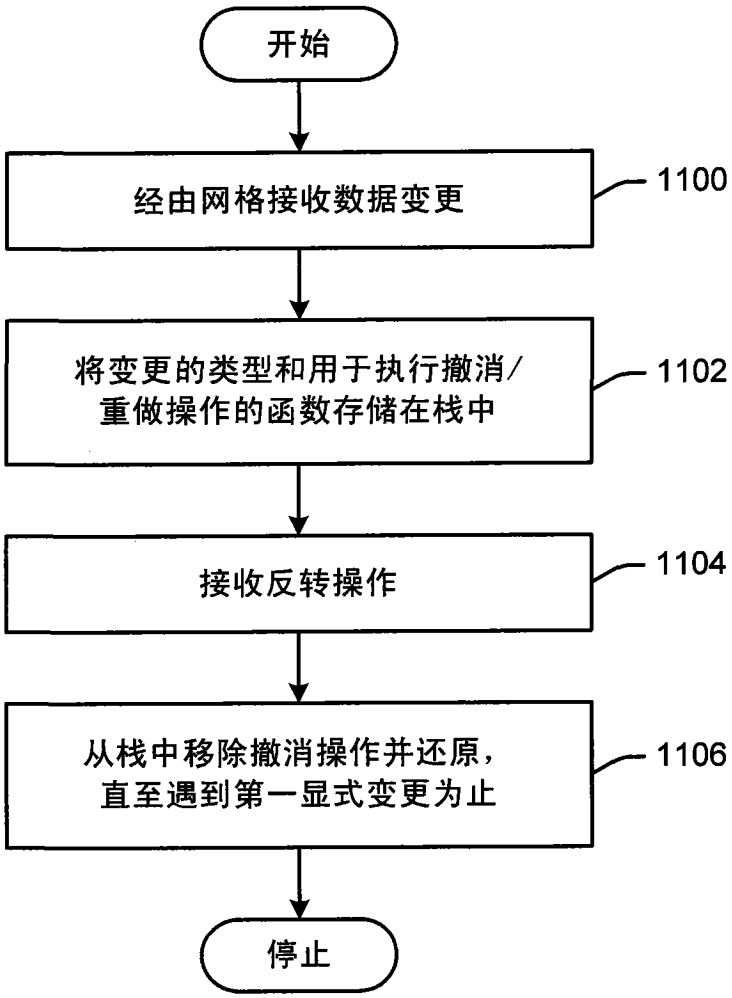

开始
经由网格接收数据变更
1100
将变更的类型和用于执行撤消/
重做操作的函数存储在栈中
1102
接收反转操作
1104
从栈中移除撤消操作并还原，
直至遇到第一显式变更为止
1106
停止

图 11

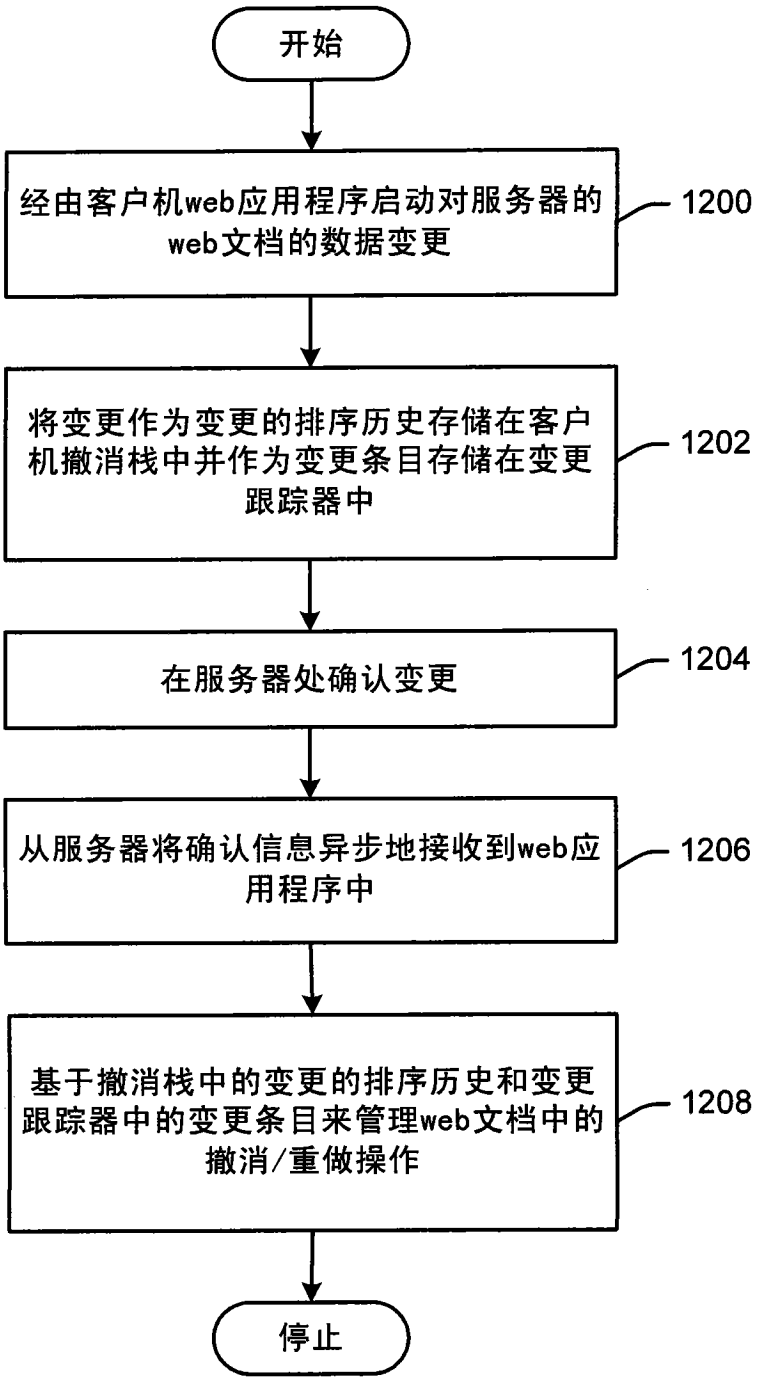

开始
经由客户机web应用程序启动对服务器的web文档的数据变更
1200
将变更作为变更的排序历史存储在客户机撤消栈中并作为变更条目存储在变更跟踪器中
1202
在服务器处确认变更
1204
从服务器将确认信息异步地接收到web应用程序中
1206
基于撤消栈中的变更的排序历史和变更跟踪器中的变更条目来管理web文档中的撤消/重做操作
1208
停止

图 12

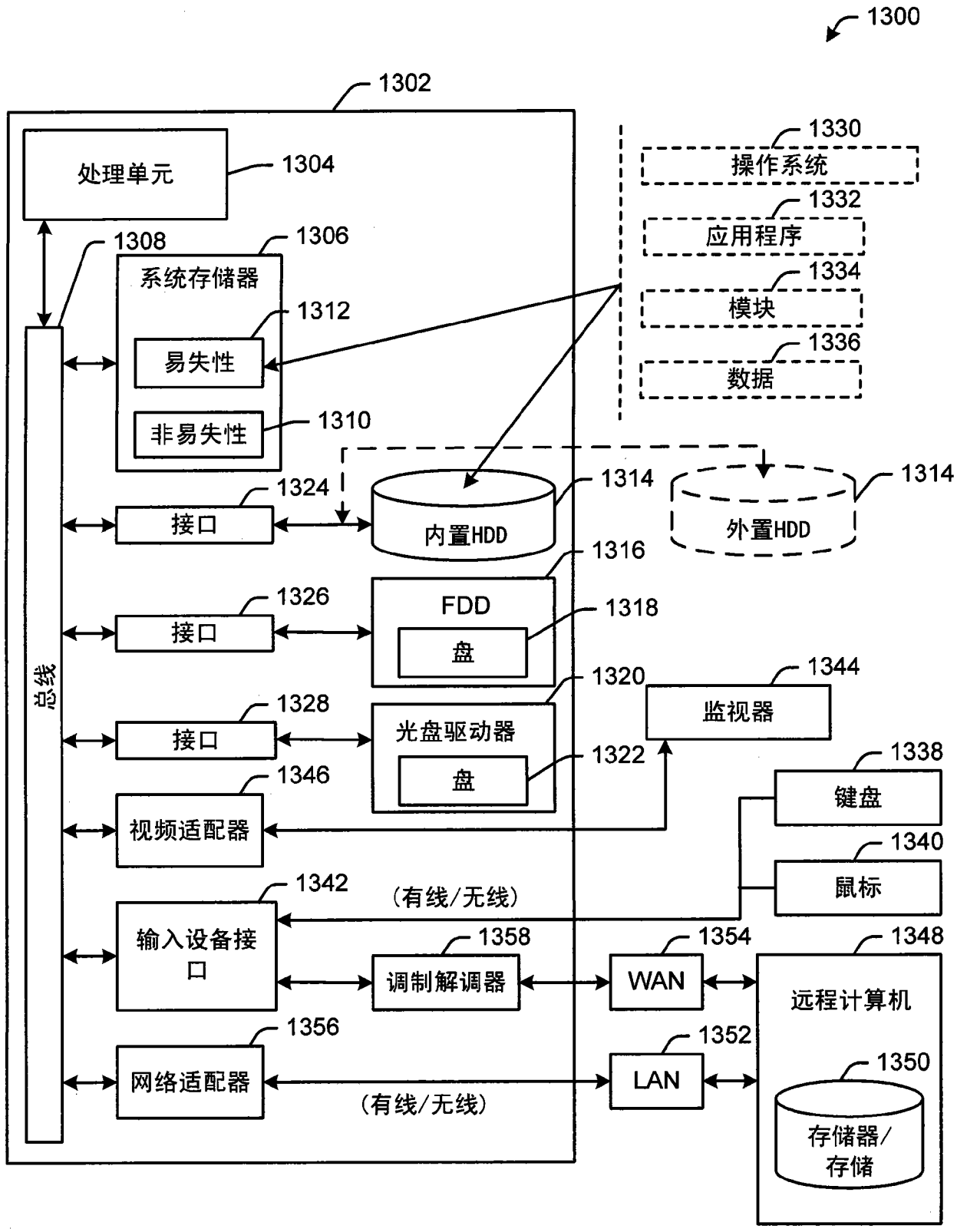

1300
1302
处理单元
1304
1308
1306
系统存储器
1312
易失性
非易失性
1310
1330
操作系统
1332
应用程序
1334
模块
1336
数据
1324
接口
内置HDD
1314
外置HDD
1314
1316
FDD
盘
1318
1326
接口
1320
光盘驱动器
盘
1322
1328
接口
1344
监视器
1338
键盘
1340
鼠标
1346
视频适配器
1342
输入设备接口
(有线/无线)
1358
调制解调器
1354
WAN
1348
远程计算机
1356
网络适配器
1352
LAN
1350
存储器/存储
(有线/无线)
总线

图 13

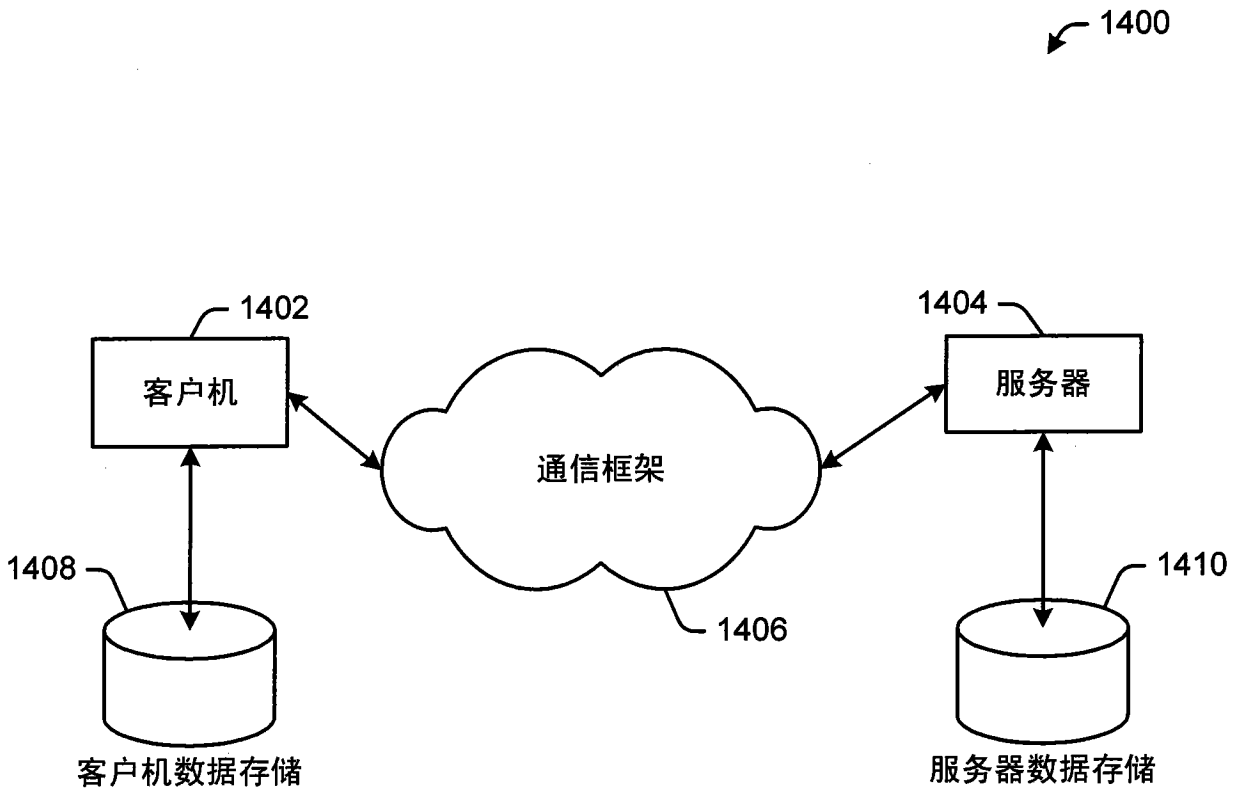



图 14