



(12)发明专利申请

(10)申请公布号 CN 110222106 A

(43)申请公布日 2019.09.10

(21)申请号 201910398862.7

G06Q 10/06(2012.01)

(22)申请日 2011.05.31

(30)优先权数据

61/351,839 2010.06.04 US

13/115,090 2011.05.24 US

(62)分案原申请数据

201180027405.9 2011.05.31

(71)申请人 MCL系统有限公司

地址 美国弗吉尼亚州

(72)发明人 斯瓦米·莫苏威路

(74)专利代理机构 成都超凡明远知识产权代理

有限公司 51258

代理人 魏彦 洪玉姬

(51)Int.Cl.

G06F 16/25(2019.01)

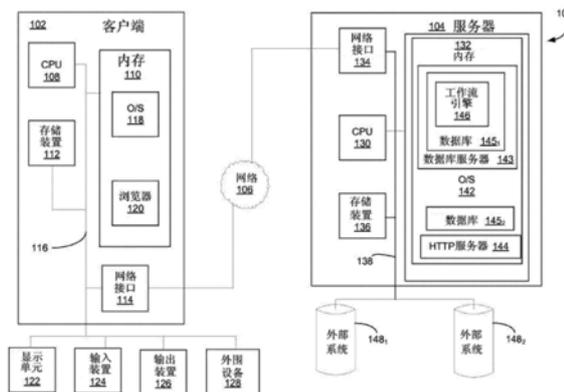
权利要求书2页 说明书10页 附图10页

(54)发明名称

集成的工作流和数据库事务

(57)摘要

本发明涉及用于集成工作流和数据库事务的计算机实施的方法和系统。还涉及包括程序的计算机可读存储媒介。计算机实施的方法包括将存储在数据库中的数据结构分派给一个或多个工作流。计算机实施的方法还包括响应正在被新记录填充的数据结构而自动地创建工作流的实例。



1. 一种计算机实施的方法,包括:

将存储在数据库中的数据结构分派给一个或多个工作流程,所述数据库包括 workflow 引擎;以及

作为正在用新记录填充数据结构的集成的数据库 workflow 事务的一部分,通过所述 workflow 引擎而创建所述一个或多个工作流程中至少一个 workflow 流程的实例,其中作为用新记录填充数据结构的一部分,所述 workflow 引擎通过一个或多个表级触发器控制数据库事务,执行 workflow 步骤以及在完成所述 workflow 事务后向所述数据库发出信号以使所述数据库事务完成或失效,和/或作为集成的数据库 workflow 删除记录事务的一部分,移除所述一个或多个工作流程中至少一个 workflow 流程的实例,其中作为删除记录事务的一部分,所述 workflow 引擎通过所述表级触发器控制数据库事务,执行 workflow 步骤以及在完成所述 workflow 事务后发出信号以使数据库事务完成。

2. 如权利要求1所述的计算机实施的方法,进一步包括:

将应用程序分派给所述一个或多个工作流程,其中该应用程序在数据结构上执行操作。

3. 如前述权利要求中的一项所述的计算机实施的方法,其中数据结构为包括多行的基表,并且其中每一行代表一个或多个工作流程的实例,并且所述基表上的数据库事务由于插入事务、更新事务或删除事务启动对应的工作流事务,但工作流事务由于 workflow 状态的变化并不会在所述基表上创建任何数据库事务。

4. 如权利要求1所述的计算机实施的方法,其中所述一个或多个工作流程受数据结构中的数据的改变控制。

5. 如权利要求3所述的计算机实施的方法,其中 workflow 事务当在数据结构中添加、更新或删除记录时通过激活所述一个或多个触发器而完成。

6. 一种包括程序的计算机可读存储媒介,所述程序在处理器上执行时执行操作,该操作包括:

将存储在数据库中的数据结构分派给一个或多个工作流程,所述数据库包括 workflow 引擎;以及

作为正在用新记录填充数据结构的集成的数据库 workflow 事务的一部分,通过所述 workflow 引擎而创建所述一个或多个工作流程中至少一个 workflow 流程的实例,所述 workflow 引擎通过表级触发器控制数据库事务,执行 workflow 步骤以及在完成所述 workflow 事务后向所述数据库发出信号以使所述数据库事务完成或失效,和/或作为集成的数据库 workflow 删除记录事务的一部分,移除所述一个或多个工作流程的实例,其中作为删除记录事务的一部分,所述 workflow 引擎通过所述表级触发器控制所述数据库事务,执行 workflow 步骤以及在完成所述 workflow 事务后向所述数据库发出信号以使所述数据库事务完成。

7. 如权利要求6所述的计算机可读存储媒介,进一步包括:

将应用程序分派给所述一个或多个工作流程,其中该应用程序在数据结构上执行操作。

8. 如权利要求6和7中的一项所述的计算机可读存储媒介,其中数据结构为包括多行的基表,并且其中每一行代表所述一个或多个工作流程中的一个或多个 workflow 流程的实例,并且所述基表上的数据库事务由于插入事务、更新事务或删除事务启动对应的工作流

事务,但 workflow 事务由于 workflow 状态的改变并不会在所述基表上创建任何数据库事务。

9. 如权利要求8所述的计算机可读存储媒介,其中 workflow 事务当在数据结构中添加、更新或删除记录时通过激活所述一个或多个触发器而完成。

10. 一种系统,包括:

处理器;和

包括数据库的内存,其中所述数据库包括 workflow 引擎层,并且其中,所述内存进一步包括程序,所述程序在由所述处理器执行时配置为执行操作,该操作包括:

将存储在数据库中的数据结构分派给一个或多个 workflow 流程;以及

作为正在将新记录填充数据结构的集成的数据库 workflow 事务的一部分,创建所述一个或多个 workflow 流程的实例,其中作为用新记录填充数据结构的一部分,所述 workflow 引擎通过表级触发器控制数据库事务,执行 workflow 步骤以及在完成所述 workflow 事务后向所述数据库发出信号以使所述数据库事务完成或失效,和/或作为集成的数据库 workflow 删除记录事务的一部分,移除所述一个或多个 workflow 流程的实例,所述 workflow 引擎通过所述表级触发器控制数据库事务,执行 workflow 步骤以及在完成所述 workflow 事务后向所述数据库发出信号以使数据库事务完成。

11. 如权利要求10所述的系统,进一步包括:

将应用程序分派给一个或多个 workflow 流程,其中该应用程序在数据结构上执行操作。

12. 如权利要求10和11中的一项所述的系统,其中数据结构为包括多行的基表,并且其中每一行代表所述一个或多个 workflow 流程中的一个或多个 workflow 流程的实例,并且所述基表上的数据库事务由于插入事务、更新事务或删除事务启动对应的工作流事务,但 workflow 事务由于 workflow 状态的改变并不会在所述基表上创建任何数据库事务。

13. 如权利要求12所述的系统,其中 workflow 事务当在数据结构中添加、更新或删除记录时通过激活所述一个或多个触发器而完成。

14. 如权利要求10所述的系统,其中至少一个数据库前端应用工具在不知晓存在 workflow 引擎的情况下操作。

15. 如权利要求10所述的系统,其中应用程序和数据库前端应用工具中的至少一个在知晓 workflow 引擎的情况下利用 workflow 引擎应用程序界面(API)操作。

16. 如权利要求10所述的系统,其中 workflow 引擎创建一个或多个事件类型,其中所述一个或多个事件类型中的每一个代表 workflow 运动类型,其中每一个 workflow 运动类型与一个或多个 workflow 元素关联,其中所述一个或多个 workflow 元素中的每一个执行与至少一个或多个数据库关联的程序代码,由此提供集成的 workflow 和数据库事务。

17. 如权利要求16所述的系统,其中内存进一步包括面向服务的体系结构(SOA)或服务代理中的至少一个,所述面向服务的体系结构或服务代理启用集成的数据库延迟的 workflow 事务。

## 集成的工作流和数据库事务

### 技术领域

[0001] 本发明大体上涉及商务过程管理(“BPM”)和工作流管理、计算机软件工业和关系数据库技术的领域。

### 背景技术

[0002] 大多数商务应用程序运行在诸如Oracle DB2或Microsoft SQL Server的关系型数据库上。商务过程管理(“BPM”)、也已知为工作流管理、是被组织机构采用以规划面向过程的方法从而完成任务的一种迅速发展的概念。实施BPM解决方案的若干软件解决方案作为单独的程序工作,或者通过使用应用程序界面(API)、命令行界面(CLI)和其他界面在n层体系结构中与其他软件平台(例如数据库、操作系统和第三方软件)结合工作。许多BPM软件解决方案从文档、图像和邮件管理系统演变而来,所有这些管理系统都涉及这样的概念,即当创建一个主题(“文档”或“文件”或“邮件”或“记录”)时,一个流程从始至终控制该主题。对于集成数据库和工作流解决方案的尝试涉及松耦合不同软件平台作为针对特定商务需求的客户解决方案。

### 发明内容

[0003] 需要理解的是:以下总结和详细说明是示例性和解释性的,致力于提供对如所声明那样的本发明的进一步解释。以下总结和说明并不意欲将本发明的范围限定或限制成在该总结或说明中提及的特定特征。

[0004] 在特定实施例中,所公开的实施例可包括本文所描述的一个或多个特征。

[0005] 在本发明的一个实施例中,提供一种计算机实施的方法。该计算机实施的方法包括将存储在数据库中的数据结构分派给一个或多个工作流流程。该计算机实施的方法还包括响应正在被新记录填充的数据结构而自动地创建工作流的实例。

[0006] 在本发明的另一实施例中,提供一种包括程序的计算机可读存储媒介,所述程序在处理器上执行时执行一个操作。该操作包括将存储在数据库中的数据结构分派给一个或多个工作流流程。该操作还包括响应正在被新记录填充的数据结构而自动地创建工作流的实例。

[0007] 在本发明的又一实施例中,提供一种系统。该系统可包括:处理器和包括程序的内存,所述程序在由处理器执行时配置为执行一个操作。该操作包括将存储在数据库中的数据结构分派给一个或多个工作流流程。该操作还包括响应正在被新记录填充的数据结构而自动地创建工作流的实例。

[0008] 本发明的这些和其他目的和特征在包括以上和以下描述的公开内容中显而易见。

### 附图说明

[0009] 可参照以下结合附图的描述理解本发明,在附图中,附图标记最左侧的标示数字表示出现相应附图标记的第一个图,并且在附图中:

- [0010] 图1示出根据本发明的一个实施例的例示出示例性计算环境的框图；
- [0011] 图2示出根据本发明的一个实施例的工作流引擎的示例性实施方案的框图；
- [0012] 图3例示出根据本发明的一个实施例的示例性基表和子表的框图；
- [0013] 图4例示出根据本发明的一个实施例的用于设计和管理工作流程的图形用户界面；并且
- [0014] 图5例示出根据本发明的一个实施例的两个表的框图，其示出数据库事务和工作流事务之间的关系；
- [0015] 图6例示出根据本发明的一个实施例的用于数据库中的工作流引擎启用和创建流程设计的流程图；
- [0016] 图7例示出根据本发明的一个实施例的用于执行数据库-工作流集成事务的流程图；
- [0017] 图8例示出根据本发明的一个实施例的用于执行工作流唯一事务的流程图；
- [0018] 图9例示出根据本发明的一个实施例的用于执行与工作流事务关联的工作流事件的流程图；
- [0019] 图10例示出根据本发明的一个实施例的用于使用工作流事务中的服务代理执行工作流事件的流程图。
- [0020] 本发明可具有各种修订方案和可替代形式，其特定实施例已作为实例示于附图中且在本文详细描述。然而，应该理解，这些特定实施例的描述并不意欲将本发明限制于特定形式，而是试图覆盖由所附权利要求限定的本发明的精神和范围内的所有修订方案、等同替换和可替代方案。

### 具体实施方式

[0021] 以下描述本发明的示例性实施例。为了清楚，在本说明书中并未描述实际实施方案的所有特征。当然将意识到，在任意这些实际实施方案的扩展方案中，可作出多种使实施方案特定化的决定以实现开发者的特定目标，例如与各种实施方案的彼此不同的系统相关和商务相关的约束条件相妥协。而且，将意识到，这样的扩展尝试可能很复杂且耗时，但对于能从本公开内容受益的本领域普通技术人员而言可以是例行公事而已。

[0022] 现在将参照附图描述本发明的各实施例。仅出于解释的目的在附图中示例性地描述各个结构、连接、系统和装置，以不会使所公开的主题与本领域技术人员已知的细节相混淆。然而，所包含的附图用于描述和解释本发明的示例性示例。本文所使用的词语和习语应理解和解释为具有与相关领域技术人员对这些词语和习语的理解一致的含义。术语或习语的特殊定义，即不同于本领域技术人员所理解的普通和常规含义的定义，并不意欲被本文的术语或习语的一致使用所掩盖。当术语或习语具有特殊含义，即与技术人员所理解的含义不同的含义时，这种特殊定义将以直接且明确地提供针对该术语或习语的特殊定义的定义方式在说明书中清楚地阐释。

[0023] 本发明的一个实施例实施为程序产品，其使用于计算机系统，例如图1所示的在下文描述的计算环境100。该程序产品的程序定义实施例(包括本文所述的方法)的功能，并可包含在各种信号承载媒介上。示例性的信号承载媒介包括但不限于：(i) 永久地存储在非可写存储媒介(在计算机内的只读记忆装置，例如由CD-ROM驱动器可读的CD-ROM盘)的信息；

(ii) 存储在可写存储媒介(例如,在磁盘驱动器或硬盘驱动器内的软盘);和(iii)由通信媒介、例如通过计算机或手机网络(包括无线通信)、传递给计算机的信息。第(iii)种实施例特别是包括从互联网和其他网络下载的信息。这种信号承载媒介在携带指示本发明功能的计算机可读指令时代表本发明的实施例。

[0024] 总之,执行实施本发明实施例的例程序可为操作系统或特定应用程序、数据库、组件、程序、模型、对象或指令系列的一部分。本发明的计算机程序典型地包括多种指令,这些指令将由本地计算机翻译成机器可读的形式并因而成为可执行指令。另外,程序包括变量和数据结构,这些变量和数据结构或者位于本地程序中,或者位于内存或存储装置中。此外,以下描述的各种程序可基于其在本发明特定实施例中实施的应用程序被标识。然而,应该意识到,以下任何特殊的程序术语仅出于方便而使用,因而本发明不应限制于仅在由这种术语所认同和/或暗示的特定应用程序中使用。就此而言,对于特殊定义语言和程序语言(例如HTML、XML、SQL、.NET、C#等)的参照仅为示例性的。可广义地考虑到,可与用于定义的网络资源内容的特殊模式或语言无关地应用本发明。

[0025] 现在翻到图1,示出根据本发明实施例的例示出示例性计算环境100的框图。大体而言,计算环境100包括客户端(例如用户端)计算机102和服务器计算机104。客户端计算机102和服务器计算机104可为相同计算机系统的组件或可经由网络106(例如互联网)连接。

[0026] 如所示,客户端计算机102包括中心处理单元(CPU)108,其经由总线116连接到内存110、存储装置112和网络接口114。所包括的CPU 108代表单CPU、多CPU、具有多个处理核的单CPU等。存储装置112存储由客户端计算机102使用的应用程序和数据。存储装置112的示例包括一个或多个硬盘驱动器、闪存装置、光学媒介等。客户端计算机102可使用网络接口114连接到通信网络106(例如,局域网,其自身可连接到诸如互联网的其他网络)。内存110可为一个或一组记忆装置,包括随机存取存储器、非易失或后备存储器(例如,可编程存储器或闪存、只读存储器,等)。示例性地,客户端计算机102的内存110存储用于管理在客户端计算机102上执行的硬件和软件的操作系统118。如所示,内存110还包括浏览器程序120,其在由CPU108执行时为在各个服务器与一个或多个服务器(例如服务器计算机104)的局域网地址之间的导航提供支持。

[0027] 客户端计算机102可连接到一个或多个显示单元122、输入装置120、输出装置126和外围设备128。显示单元122可为内部或外部的监视器、电视屏幕、手持设备显示器等。输入装置124可为键盘、鼠标、轨迹球、定位笔、鼠标垫、鼠标按钮、操作杆、扫描器等中的任意一个。输出装置126可为监视器、打印机、绘图机、复印机或其他输出装置中的任意一个。外围设备128可为能够联接到计算机的任何其他装置:能够读和/或写到物理数字媒介上的CD/DVD驱动器、USB装置、极碟驱动器、外部软盘驱动器、外部硬盘驱动器、电话和/或宽带调制解调器、路由器/网关、访问点等。

[0028] 类似于客户端计算机102,服务器计算机104可包括经由总线138联接的CPU 130、内存132、网络接口装置134和存储装置136。内存132可为随机存取存储器,其大到足以容纳位于服务器计算机104上的必要的编程和数据结构。如所示,内存132存储用于管理在服务器计算机104上执行的服务器硬件和软件的操作系统142。示例性地,内存132还包括超文本传输协议(http)服务器144,其配置为维护来自客户端计算机102的请求。例如,http服务器144可响应请求以访问位于服务器计算机104上的电子资源(例如,HTML文档、网络信息等)。

内存132也可包括数据库服务器143,其可响应请求以访问数据库应用查询等。然而,本领域普通技术人员将意识到,数据库服务器143或http服务器144仅为示例性的,并且本发明的实施例可适于支持已知和未知的协议。http服务器144的编程和数据结构可根据需要在操作期间由CPU 130存取和执行。服务器计算机104可利用网络接口装置134(例如,模拟调制解调器、有限网卡或无线网络装置)连接到网络106。

[0029] 在一个实施例中,用户可利用图形用户界面(GUI)与服务器计算机104交互。在特定实施例中,GUI内容可包括HTML文件(即,网页),其利用浏览器120呈现在与客户端计算机102联接的显示单元122上。在一个实施例中,网页可包括允许用户设计、操纵、执行和监视各个商务流程(即 workflow)的执行情况的页面。

[0030] 数据库服务器143可包括多个数据库145<sub>1</sub>、145<sub>2</sub>。数据库145<sub>1</sub>可作为 workflow引擎146的主机。 workflow引擎146可包括软件应用程序,其配置为能够(例如,经由GUI)模型化 workflow(例如,定义流程、状态、任务、动作、事件、规则等),以控制与一个或多个数据库关联的活动。数据库145<sub>1</sub>、145<sub>2</sub>可包括使用SQL查询询问的关系型数据库145<sub>1</sub>或使用XML查询询问的XML数据库145<sub>2</sub>。然而,本发明不限于任何特定的物理数据库存储机制,并可易于扩展以在其他当前已知或未知的这种机制上操作。尽管数据库145<sub>1</sub>、145<sub>2</sub>示例性地位于服务器系统内部,但可注意到,数据库145<sub>1</sub>、145<sub>2</sub>可存在于服务器计算机104的外部存储装置(例如,存储装置136)上,或者可经由网络106存取。

[0031] 现在翻到图2,例示出根据本发明实施例的 workflow引擎146的示例性实施方案的图表。 workflow引擎146位于数据库145<sub>1</sub>中。数据库145<sub>1</sub>位于数据库服务器143中。如所示, workflow流程设计器204、商务软件应用程序206、数据库查询工具207、数据库报告工具208和其他数据库工具209联接到数据库145<sub>1</sub>。在一个实施例中,数据库服务器143可为数据库管理系统(DBMS),其控制数据库145<sub>1</sub>、145<sub>2</sub>的创建、维护和使用。数据库145<sub>1</sub>、145<sub>2</sub>与逻辑单元相关,在该逻辑单元中,数据、过程和安全独立地由一个或多个相关的应用程序使用。数据库服务器143也可联接到外部系统148<sub>1</sub>、148<sub>2</sub>,其可为第三方系统,该第三方系统通过利用数据库145<sub>1</sub>内的可编程特性和/或利用在数据库服务器143中可用的软件特性或在O/S 142中(如图1所示)可用的特性由 workflow引擎146直接控制或编排。在一个实施例中,在许多本领域已知的领先的数据库服务器中可用的服务代理特性(未示出)用于编排。

[0032] 可注意到,编排涉及自动安排、协调和管理复杂的计算机系统、中间设备和服务。在编排操作中,消息在松散连接的系统之间交换,这些松散连接的系统共同工作以完成任务。作为示例,第三方付货公司可被发送消息以将产品运送到一个地址。付货公司独立地执行运送并向请求者确认付货细节。请求付货的公司并不关心付货物流,并且付货公司并不关心付货目的。消息简单地在两个公司系统之间交换以独立地完成付货任务。

[0033] 流程设计器204允许用户(例如,经由浏览器120(在图1中示出)呈现的GUI)创建和修改 workflow流程。利用GUI,用户可定义、编辑、回顾和管理 workflow流程。当 workflow流程被流程设计器204创建时,数据库145<sub>1</sub>中的表(或数据结构)被定义为用于流程的控制表(“基表”)。数据向基表中的流入触发 workflow引擎146。流入基表(表的记录)的数据的实例启动 workflow的实例。基表可为用于多于一个流程的控制表,因此流入基表的实例可启动针对每一个流程的 workflow的许多实例。换言之, workflow事务作为数据库事务到基表的一部分而启动。

[0034] 大体上, workflow包括三个基本元素:流程、状态和转换。 workflow流程为用于完成任

务的工作流动的序列。根据由流程设计器204设定的条件,该流动可在不同的点采取不同的路由。大型任务可逻辑性地划分为许多相互连接的子流程。状态为在工作流流程中的静止步骤。转换为将工作流实例从一个状态移动到下一个状态的动作。转换为特定用户动作,或者为基于由用户设定的特定条件的自动流动。利用GUI,用户可定义正在模型化的工作流的状态和关联的转换。在一个实施例中,工作流可模型化为高级流程图(在图4中示出),其使用户能够逻辑性地描述作为流程的活动。在一个实施例中,用户可通过执行图标的“拖拽”操作来创建流程图。流程设计器204与工作流引擎146交互。所有工作流流程定义(“工作流元数据”)作为数据由数据库145<sub>1</sub>存储。

[0035] 应用程序206为用于处理软件所被设计用于支持的商务的商务应用程序。作为示例,用于邮购操作的客户订单管理应用程序的软件为商务应用程序。其可具有用于创建订单、付货订单和支付客户的屏面。这种应用程序可被建立为具有工作流能力或没有工作流能力。非工作流应用程序可在如145<sub>2</sub>中的数据库内不具有工作流引擎146。具备工作流能力的程序如同任何软件应用程序那样工作,但具有额外的工作流引擎146的控制。该应用程序可被改进为工作流感知的应用程序,此时应用程序206与工作流引擎146交互作用并利用工作流特性。作为示例,工作流被改进的应用程序可显示所有开放订单的工作流状态。工作流被改进的应用程序可向用户委派和优先化工作。工作流被改进的应用程序还可控制哪些用户可做哪些工作并在用户之间进行协调等。

[0036] 如前所述,工作流引擎146位于数据库145<sub>1</sub>中并与数据库145<sub>1</sub>和数据库服务器143集成。该集成通过将工作流引擎146实施为位于数据库145<sub>1</sub>的数据库层中的程序而实现。这种集成使工作流引擎146具有当新数据输入数据库145<sub>1</sub>中时或当数据被修改和/或删除时控制工作流流程的能力。该集成提供对数据库功能的透明式使用(即,不知晓工作流引擎146的存在)。然而,该集成包括工作流控制。工作流流程在对位于数据库145<sub>1</sub>中的基表中的数据的改变下被实例化、引导和控制。通过应用程序206、数据库查询工具207、数据库报告工具208和许多其他数据库工具209对数据库145<sub>1</sub>中的数据的任何改变由工作流引擎146自动探测和控制。该集成提供对数据库145<sub>1</sub>层内的工作流引擎146的无缝实现。

[0037] 工作流引擎146可提供工作流流程的创建和管理、解译工作流的模型和管理工作流执行。工作流引擎146也可为流动设计路由和使各活动同步、向各活动分派资源、通知个体、调用事件和应用程序、传送数据和文件给应用程序和个体、控制用户安全和特权、维护各个工作流实例的各个状态的信息、和维护工作流事务的日志。

[0038] 现在翻到图3,示出根据本发明实施例的示例性基表和子表的框图。大体上,用于商务、组织、实体等的数据库145<sub>1</sub>、145<sub>2</sub>中的若干关系表302、304、306中。一个这种表为基表302,其典型地包含多行,每行包含记录。如果把基表放置在订单管理应用程序的环境下,基表中的每个记录可代表使用应用程序下的订单。如所示,每个记录可包含每个所下订单的订单识别号。基表302可具有关联的子表302、304的N个数。每个子表302、304可包含关于每个所下订单的信息。如所示,子表304包含每个所下订单的客户的名称,子表306包含每个订单购买的物品的数量。在一个实施例中,一旦基表302被新纪录填充,则工作流被启动。因此,基表302的每行也可代表工作流流程的发生(例如如由工作流定义的订单管理流程)。应用程序206可配置为创建或修改存储在基表302(或其子表304、306中的一个)内的数据。应用程序206可使用用户界面输入或修改数据、提供报告能力等。

[0039] workflow也可与内部和外部系统148<sub>1</sub>和148<sub>2</sub>关联。例如,考虑这样的商务模型,其中客户被信任为具有购买的频繁买家点。在该情况下,被信任为具有点的客户频繁买家点系统可为内部系统。然而,其为使用数据库145<sub>2</sub>的不同应用程序。在另一示例中,第三方付货公司可用于客户订单的直接装运。付货公司系统(例如,外部系统148<sub>1</sub>和/或148<sub>2</sub>)可被通知付货细节。付货公司可执行付货任务和发送信息细节。与内部和外部系统的交互作用通过workflow事件完成。workflow事件可使用可编程语言(例如,C#、Java等)在脚本中规定。事件脚本可作为数据库workflow集成事务的一部分或者只作为workflow事务的一部分由workflow引擎146执行。事件可被设定为在适当的地点执行,例如当进入状态、离开状态时或者当用户执行动作时执行。事件包括但不限于:创建文件、启动其他系统和/或应用程序中的动作、发送邮件通知等。

[0040] 现在翻到图4,例示出根据本发明实施例的用于设计和管理workflow流程404的图形用户界面屏面402。出于示例的目的,所示的workflow流程404用于订单管理流程。如所示,订单管理流程可包括多个状态:开始406、在制订单408、付货订单410、开票订单412和结束414。状态406、408、410、412、414可通过多个转换而彼此联接:下订单416、付货418、开票420、取消422和完成424。workflow流程404可配置为使得当购买者(例如经由网上表格)对产品下订单时其进入在制订单状态408。根据在该状态期间采取的动作,workflow404可从在制订单状态408转换到结束状态414或付货订单状态410。例如,如果订单(例如被供应商)取消,则workflow流程404可从在制订单状态408经由取消转换422转换到结束状态414。在该情况下,供应商可使用用户表格,该用户表格在订单取消时启动使workflow遵从取消转换422的事件。然而,如果产品已付货,则workflow流程404可从在制订单状态408转换到付货订单状态410。在该情况下,用户表格可启动交互作用,该交互作用在订单已付货时使workflow执行付货转换418。在(例如使用用户表格)对订单开票时,workflow流程404可经由开票转换420从付货订单状态410转换到开票订单状态412。在完成订单时,开票订单状态可经由完成转换424转换到结束状态。

[0041] 状态406、408、410、412、414和转换416、418、420、422、424可作为workflow402的一部分利用状态526中的图标和转换箱428中的图标通过拖拽状态和转换而实施。

[0042] 如前所述,workflow流程404与存储在数据库145<sub>1</sub>、145<sub>2</sub>中的基表中的数据关联。在一个实施例中,用户可利用选择数据库按钮430选择适当的数据库145<sub>1</sub>、145<sub>2</sub>。利用以上示例,用于订单管理流程的workflow流程404可使用选择基表按钮432来创建,该选择基表按钮关联包含订单信息(例如,订单标识号)的基表302(在图3中示出)。订单(workflow实例)可通过各种工具创建,这些工具包括订单处理应用程序206、查询工具或任何其他兼容的应用程序。workflow实例与使用哪种工具无关地被创建。

[0043] 在一个实施例中,在基表302关联到workflow流程404期间,workflow引擎146在基表上创建表级触发器。这种触发器用作向workflow引擎146的网关(gateways)。当在基表中添加、更新和删除记录时,workflow引擎146通过表触发器得到控制。当workflow引擎146获得控制时,其使用记录的主键标识记录并完成workflow事务。因此,workflow事务作为数据库事务的一部分被完成。一旦基表302被新订单的记录填充,则workflow流程404的实例可针对该订单自动创建。在该情况下,当workflow流程404被启动时,一旦基表302被新下的订单插入,其可配置为自动地和/或有条件地从开始状态406转换到在制订单状态408。

[0044] 一旦用户设计所希望的工作流流程404,则 workflow 能力可并入现有的或新的应用程序(例如,基于网页的下订单应用程序)。例如,用户可(例如,使用Microsoft Visual Basic.NET)创建网上表格并将该网上表格与数据库145<sub>1</sub>、145<sub>2</sub>关联。利用浏览器122可创建多个网上表格,其每一个均具有不同的目的。例如,一个特殊的网上表格可用于传送产品的订单。这样,网上表格可包含接收关于订单的信息(例如,付货地址、付款地址、产品名称、数量等)的各个字段。当用于传送表格时,代表新下的订单的输入在为所创建的工作流流程404选择的相同的基表302中创建。因此,一旦在基表302中创建输入,workflow 实例可针对该订单自动创建。workflow 实例可以或不可以基于针对该流程设定的条件而创建。作为示例,该条件可被设定为仅针对邮件订单且不针对店内购买的订单创建 workflow。当创建一行时,下订单转换也可被启动。因此,开始状态406可转换到在制订单状态408。workflow 的实例通过基表的主键或唯一键标识。表的主键为用于表的一行的唯一标识器。workflow 引擎146独立地维护在其自身的包括多个表的数据存储中的基表中的每个记录的信息。例如,workflow 引擎146具有ORDER\_ID=0001(主键)现在处于状态408的信息。

[0045] 可创建的另一类型的表格为订单实现和付货表格,其可被产品的供应商使用。供应商可使用该表格取消订单,或者实现所订的一个或多个产品(即,付货订单)。在提交时,该表格可启动一事件,该事件通知 workflow 引擎146将 workflow 流程404从在制订单状态408移动到适当的状态。例如,如果供应商取消订单,则可启动取消转换事件,其导致 workflow 流程404从在制订单状态408移动到结束状态414。在一个实施例中,在提交订单实现和付货表格时,可调用一功能,其发出执行取消转换422的信号,该取消转换引导 workflow 引擎146使 workflow 404从在制订单408移动到结束状态414。另一方面,如果供应商实现订单,则付货转换418可被启动,其导致 workflow 流程404从在制订单状态408移动到付货订单状态410。在一个实施例中,在提交订单实现和付货表格时,可调用一功能,其引导 workflow 引擎146将 workflow 流程404从在制订单状态408移动到付货订单状态410。

[0046] 在基表中的记录和在基表中的数据独立于 workflow 引擎146。在一个实施例中,workflow 引擎146不控制哪些数据可输入基表中。在该情况下,workflow 引擎146使用主键独立地引导和监视基表的记录的状态信息。该状态信息单独地存储在工作流引擎146的数据存储内。通过将 workflow 数据与基表数据分离,应用程序206和数据库145<sub>1</sub>独立于且透明于 workflow 地运作。但是,workflow 的完整功能由 workflow 引擎146提供。

[0047] workflow 引擎146在当前位于流程(408、410和412)中的任意状态下的基表的记录上处于活动状态。开始状态406和结束状态414是过渡状态,并且基表记录不可以在该状态下“休息”。当基表记录到达结束状态414时,该记录从 workflow 队列中移除,workflow 引擎146停止追踪该记录。在到达结束状态时,订单完成(或取消),对于该订单不再需要进一步的工作。基表继续具有关于完成订单的数据。基表数据在应用程序206的控制下保持。在另一实施例中,也可以使用基表的主键和记录应被安置的当前状态通过向 workflow 引擎146“恢复”请求而将完成的记录恢复到 workflow 中。

[0048] 当基表记录的数据通过应用程序206或查询工具207或其他数据库工具209更新时,对除了主键以外的记录数据的改变可不影响 workflow 实例。在一个实施例中,当主键使用在工作流引擎中以追踪当前状态时,如果记录在工作流中处于活动状态(例如,如果记录的当前状态为408或410或412),则 workflow 引擎206可不具有更新主键的能力。在另一实施例

中, workflow引擎146自动地调节 workflow实例键以与改变的数据库主键匹配, 由此允许对主键更新。与所使用的方法无关, 记录可在没有限制的情况下更新, 从而利用 workflow引擎146的有限干预而完全独立于应用程序206。

[0049] 当基表记录通过应用程序206或查询工具207或其他数据库工具209删除时, 如果其在工作流中保持活动(例如, 如果记录的当前状态为408或410或412), 则 workflow引擎146检查记录的主键并还将其从 workflow队列中删除。

[0050] 基表插入启动 workflow实例和事务。然而, 在不改变基表记录的情况下, 可以存在许多后续的工作流事务。工作流事务在存在从一个状态向另一状态的改变时发生。

[0051] 以上描述总结在图5中。现在翻到图5, 示出两个表502、504的框图, 这两个表显示数据库与 workflow事务之间的关系。表502显示数据库事务对 workflow事务的作用。表504显示 workflow事务对数据库事务的作用。如表502所示, 在基表502上的数据库事务启动 workflow事务。例如, 数据库事务为插入事务, 并且插入事务在满足基表条件时在 workflow事务上执行。如果数据库事务为对基表主键的更新事务, 则数据库事务将被迫失效。然而, 如果更新事务是对非主键的更新事务, 则不存在对 workflow事务的作用。如果数据库事务为删除事务, 则对应记录的主键也从 workflow中删除。

[0052] 然而, 如表504所示, workflow事务(例如, 插入、更新或删除事务)并不在基表上创建任何数据库事务。这在不改变基于数据库的应用程序开发的体系结构的情况下有助于最小的入侵性 workflow引擎146被引入在现有的或新的应用程序内。

[0053] 在另一实施例中, 集成的数据库- workflow事务可作为数据库延迟的 workflow事务(database-deferred workflow transaction)实施。从数据库事务的角度, 当一行被插入表中时, 数据库引擎通常设定行级锁或设定表级锁。锁的实施方案在各个数据库厂商之间不同。(由 workflow引擎146创建的)触发器可非常快速地执行以维持合理的能力。触发器可快到足以允许数百甚至数千个订单并行输入。本发明的许多实施例允许与内部和外部系统148集成。这种集成可能使数据库事务变缓慢或不明确地等待来自其他系统的响应。另外, 从一个状态向另一状态的流动可被设定为使若干状态在一个循环中自动地流动, 并且许多 workflow子流程可自动地创建。这种长流动可下锁系统, 这可使整个系统变缓慢。许多数据库实施方案具有对嵌套级触发器的限制, 并可防止若干步骤的长期嵌套自动流动。这些问题大体上可通过良好的 workflow流程设计解决。参见表502和504, 数据库- workflow集成事务可在插入和删除数据库事务中相关。大多数数据库系统提供用于这些事务的行级锁, 并且可能的是, 其他用户(例如, 其他订单事务)可被锁定或变缓慢。然而, 存在对嵌套数据库事务的限制到期的可能性以及不确定地等待涉及外部系统的事务的可能性等。这样的问题可利用数据库延迟的 workflow事务解决。

[0054] 为了实施数据库延迟的 workflow机制, workflow引擎146可使用服务代理或任何面向服务的体系结构(SOA)(未示出)。服务代理或SOA实施方案利用许多本领域已知的现代数据库可用。这提供解除任务和独立地以非锁定的方式完成任务的能力。当 workflow引擎146接收来自数据库事务的控制时, workflow引擎146可选择性地选择以向服务代理授权一些或所有 workflow事务。在向服务代理授权之后, workflow引擎146可向数据库引擎发出 workflow事务未决或完成的信号, 这则允许数据库事务被完成, 由此解除所有数据库锁。然而, workflow引擎146将 workflow事务标示为未决, 并防止基表记录被移动到其他状态。当服务代理成功地完成工

作并通知 workflow 引擎 146 时, workflow 引擎 146 为进一步的 workflow 事务解除对该记录的锁。如果服务代理没有完成工作(例如,未到达外部系统 148),则服务代理可等待并重新尝试。在一些条件下,(与该出差错的数据库-workflow 事务有关的)先前完成的数据库事务可编程地重新运行到初始数据库事务没有发生的水平。

[0055] 现在翻到图 6,示出根据本发明实施例的用于启用 workflow 的数据库和创建 workflow 流程的流程图。这些操作在步骤 602 开始,此时数据库被选择为在 workflow 下使用。在步骤 604,数据库被核对以检查其是否已经针对 workflow 被启用。如果数据库已经启用,则在步骤 608 存储在数据库中的基表被选择用于流程。这在基表上创建新的流程(例如,开始 406 状态,如图 4 所示)。然后,该流程在步骤 610 被改进为具有许多状态、转换、结束状态、条件、事件、安全等。如果在步骤 604 确定数据库没有针对 workflow 被启用,则在步骤 606 workflow 能力通过安装 workflow 引擎 146 被启用。

[0056] 现在翻到图 7,示出根据本发明实施例的用于数据库-workflow 集成的事务的流程图。图 7 为详细显示图 5 的表 502 中所示的事务的流程图。该流程图例示出可能的具有在基表上启动的数据库事务的数据库-workflow 集成的事务。流程图从如在步骤 702 的数据库事务开始,其由用于插入、更新和/或删除数据库事务的 workflow 146 检查。在步骤 704,在确定数据库事务为插入记录事务之后,确定数据库事务是否符合用于创建工作流实例的条件。如果其符合条件,则给老子实例在步骤 708 创建,并且插入记录数据库事务被发出信号以在步骤 710 由 workflow 引擎 146 处理。如果其不符合条件,则 workflow 实例不会在步骤 706 创建,并且数据库插入事务被发出信号以在步骤 710 由 workflow 引擎 146 处理。

[0057] 在步骤 712,在确定数据库事务为更新记录事务之后,数据库事务被检查以确定该更新是否涉及记录的主键。如果主键被更新,则其在步骤 714 进一步被检查以确定该记录是否具有 workflow 实例。如果 workflow 实例存在于主键,则在步骤 716 workflow 引擎 146 向数据库引擎发出信号以使事务失效。当 workflow 引擎 146 追踪具有主键的 workflow 实例时,主键不能被更新。在另一实施例中,workflow 引擎 146 自动地调节 workflow 实例以与改变的数据库主键(未在流程图中示出)匹配。如果主键字段没有被更新,或者如果记录没有在工作流中,则在步骤 718 workflow 引擎 146 向数据库发出信号以允许和完成数据库事务。

[0058] 在步骤 720,在确定数据库事务为删除记录事务之后,数据库事务被检查以确定 workflow 实例是否针对主键存在。如果其存在,则在步骤 722,workflow 引擎 146 删除 workflow 实例,接着在步骤 724 向数据库引擎发出信号以完成删除数据库事务。如果 workflow 事务不存在,则在步骤 724,workflow 引擎 146 向数据库引擎发出信号以完成删除数据库事务。

[0059] 现在翻到图 8,示出根据本发明实施例的用于 workflow 事务的流程图。图 8 为详细显示图 5 的表 504 中所示的事务的流程图。该流程图例示出可能的工作流事务,其并不由基表上的数据库事务启动。在步骤 802,workflow 实例 804 可通过使用当前不在 workflow 中的基表中的现有记录创建。这可在工作记录(例如,订单)被恢复以重做 workflow 时是可能的。作为示例,不正确执行的客户订单可被恢复到 workflow 以重新执行。在步骤 806,在插入事务期间,workflow 引擎 146 流到如由用户限定的恢复状态。

[0060] 在步骤 808,在更新事务期间,workflow 实例被更新。当更新 workflow 实例时,在用户或机器执行转换时一个状态移动到另一状态。在步骤 810,workflow 引擎 146 检查转换并确定下一状态是否为结束状态。如果确定下一状态为结束状态,则 workflow 实例在步骤 812 被删除。

workflow实例的删除不会对数据库基表记录产生影响。数据保持在基表中。在没有用于该记录的工作流实例的情况下，其发出存在工作未决的信号。如果下一状态并非结束状态，则在步骤814， workflow实例的当前状态为被下一状态更新。

[0061] 在步骤816，在删除事务期间， workflow实例被删除。当删除 workflow实例时，来自基表的记录被移除。

[0062] 现在翻到图9，示出根据本发明实施例的用于 workflow事务中的事件的流程图。在 workflow引擎146中，当订单从状态“在制订单”408通过使用转换“付货”418流动到状态“付货订单”410时，多个 workflow事件可被执行。转换“付货”418可由用户或系统启动。在示例的事件922、924、926中，这些事件代表以任意编程语言写出的编程代码以在任意内部或外部系统148上执行代码。在该流动中，“正离开”型事件922可存在并在离开状态408时得到执行。“正执行”型事件924可存在并在执行转换418时得到执行。“正输入”型事件926可存在并在输入状态410时得到执行。作为示例，事件926以同步的方式向邮件服务器940通信928，以向订单的接受者发送付货通知。当执行成功的转换时， workflow实例离开一个状态并进入另一状态；因此在该示例中的所有三个事件可以922、924和926的顺序得到执行。可能存在多种类型的可在 workflow运动时建立的事件。许多事件类型提供更大的间隔尺寸和环境。

[0063] 现在翻到图10，示出根据本发明实施例的在 workflow事务中的事件的情况下使用服务代理的流程图。除了服务代理1030用作 workflow引擎146与邮件服务器940之间的代理之外，该流动与图9相同。事件1026不同于926，因为其利用委派消息将1028连通到位于数据库145中的服务代理1030以发送邮件，而不等待该消息是否实际由邮件服务器940接收。服务代理1030连通到邮件服务器940，获得来自邮件服务器940的响应并将1030连通回 workflow引擎146。该方法的有点在于：如果邮件服务器暂时停止运行，则数据库- workflow事务的数据库事务组件可不必等待。 workflow事务可不得不等待，直到服务代理完成工作。这通过使用 workflow事件消除系统的任何所有的变缓慢或锁定。

[0064] 还应该注意到，对从本公开内容受益的本领域技术人员明显的是，本文所述实施例可具有大的应用范围，而不只是在本文详细描述的应用。

[0065] 以上公开的特定实施例仅为示例性的，对从本文教导受益的本领域技术人员明显的是，本发明可以不同但等同的方式修改和实践。而且，并不意欲限制为如本文所示的结构或设计的细节，除非在所附权利要求中描述。因此，明显的是，以上公开的特定实施例可被改变或修改，并且所有这样的变化在所要求的本发明的范围和精神内考虑到。

[0066] 因此，本文要求的保护范围在所附权利要求中阐释。

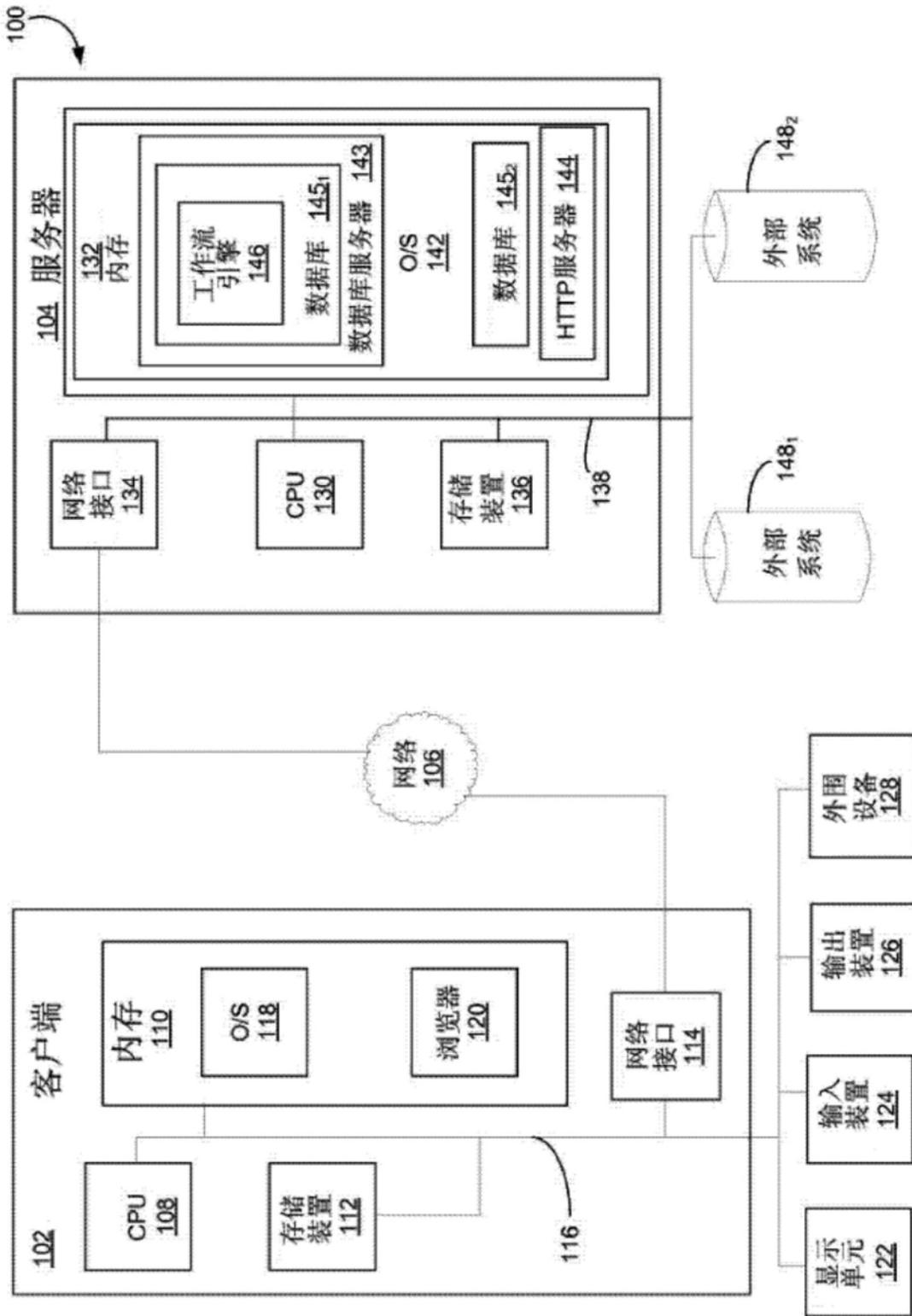


图1

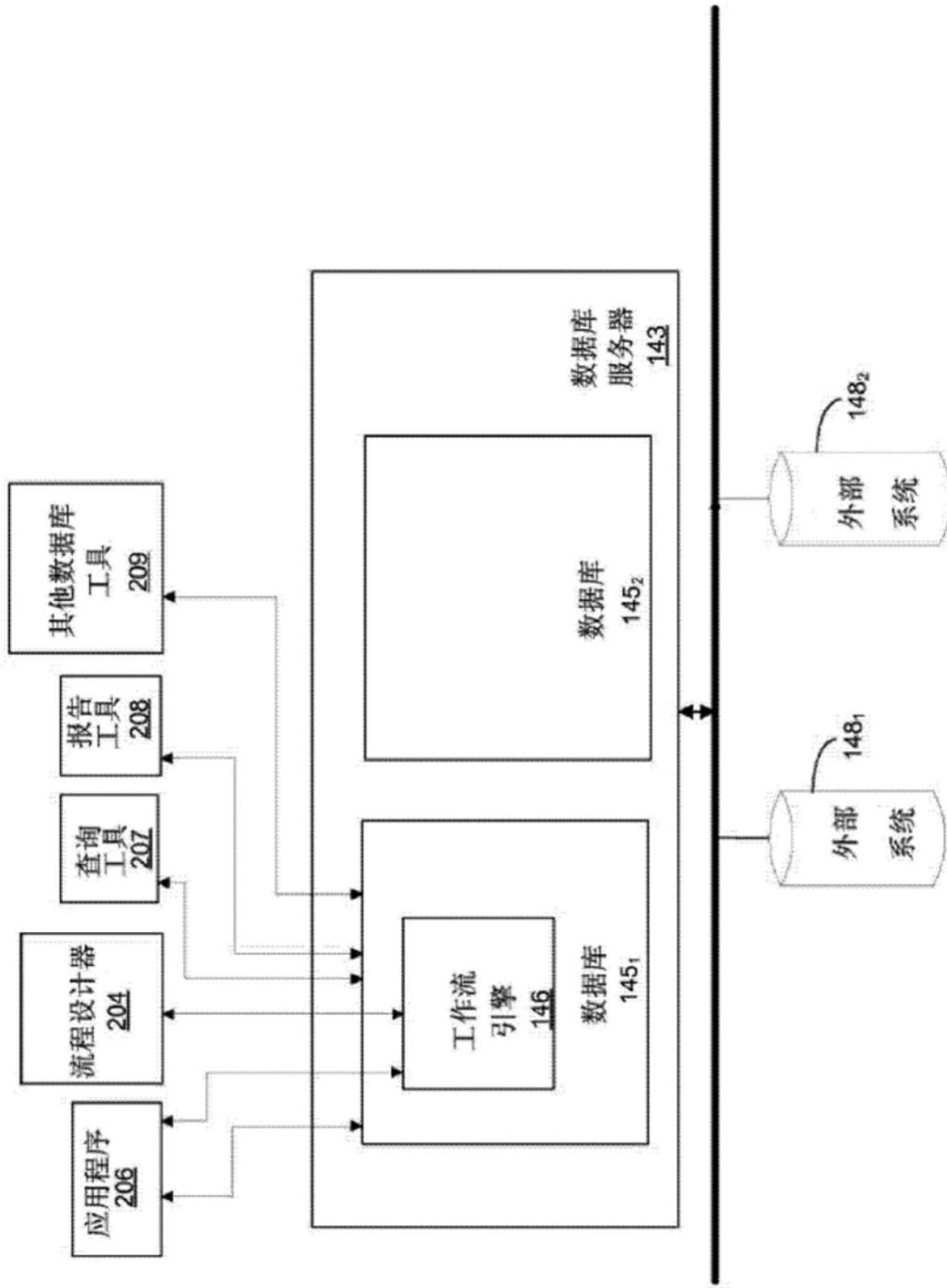


图2

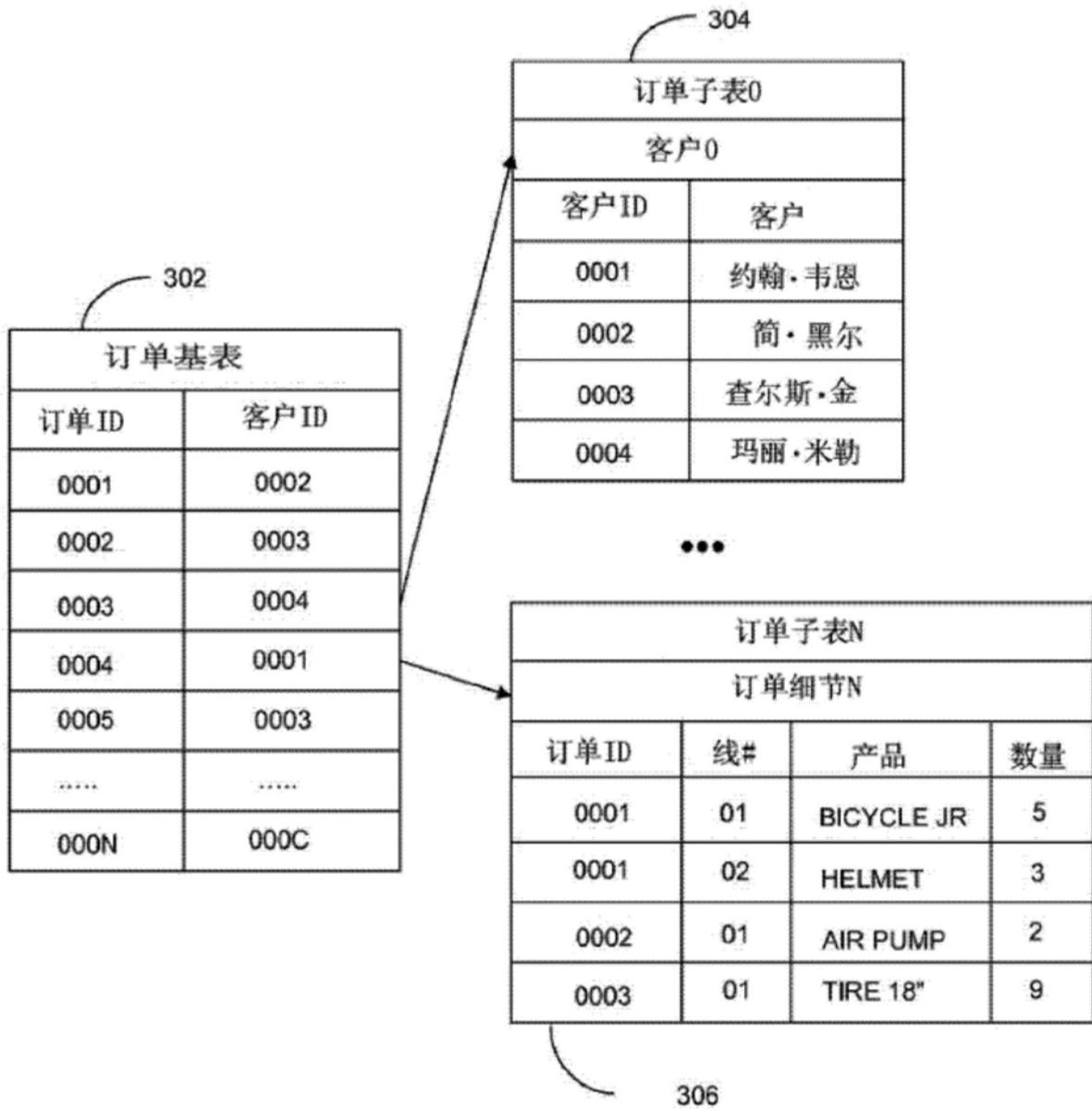


图3

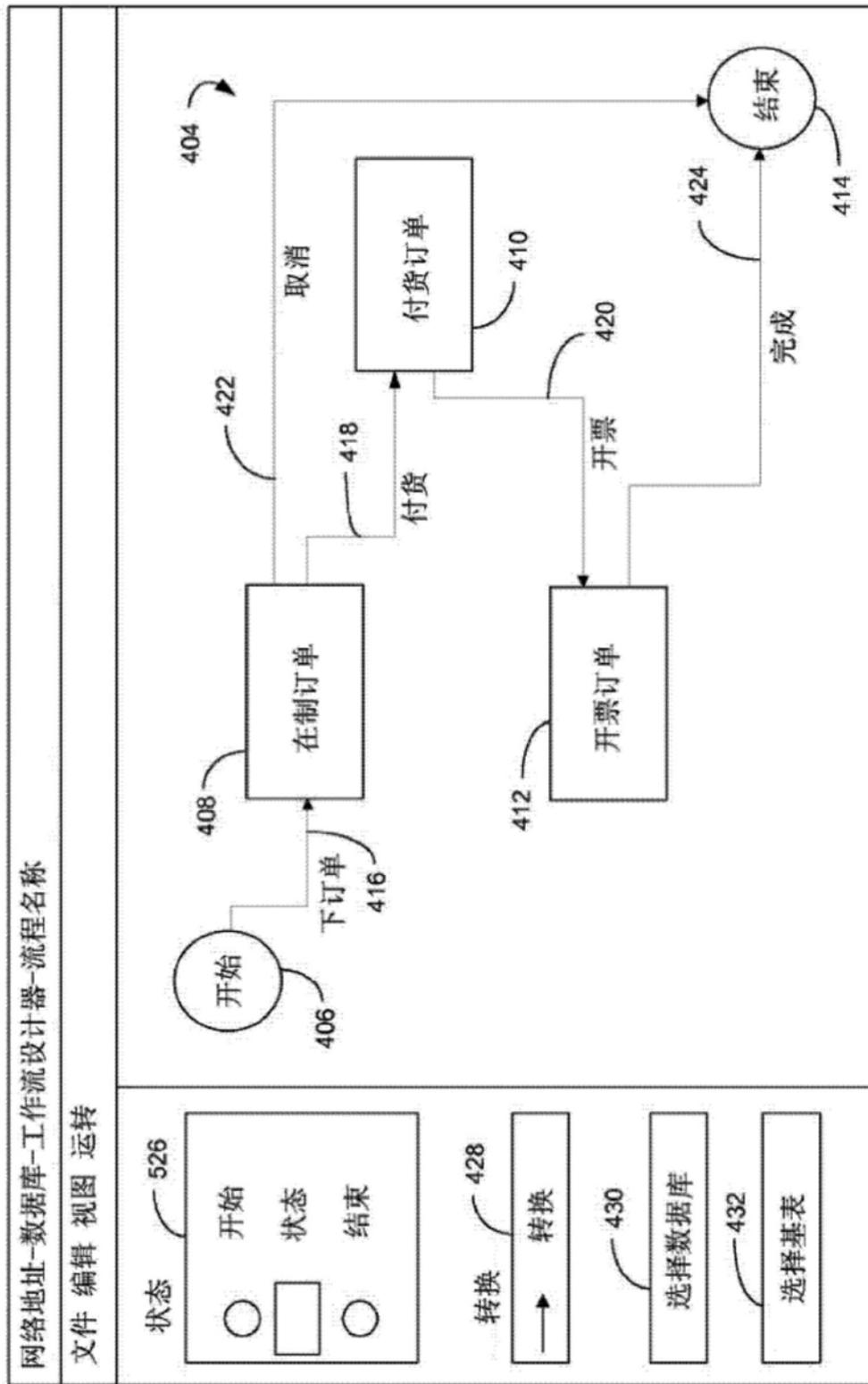


图4

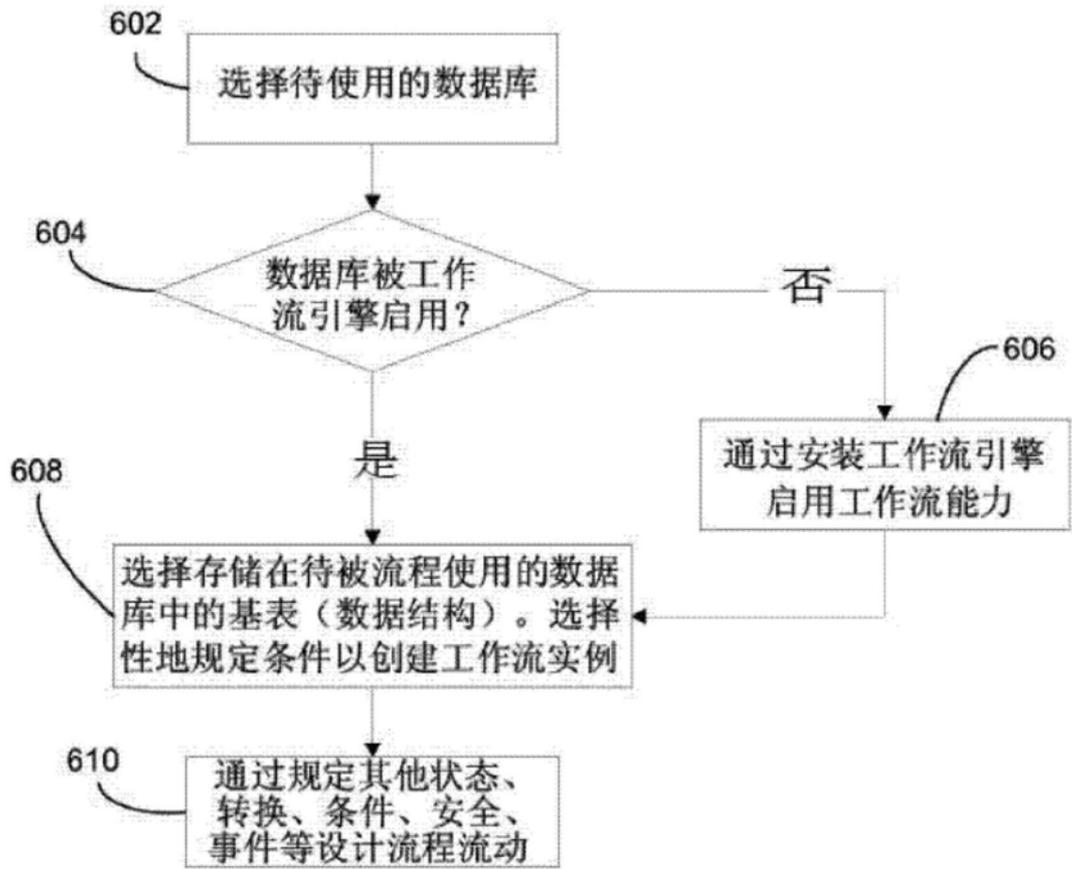
502

表502: 数据库表事务对 workflow 事务的作用	
数据库基表事务	workflow 事务
插入	插入, 如果基表条件满足 不动, 如果基表条件不满足
更新	主键的更新将迫使数据库事务失效, 如果记录在 workflow 中 非主键字段的更新不会对 workflow 事务产生影响
删除	从 workflow 中删除, 如果存在

504

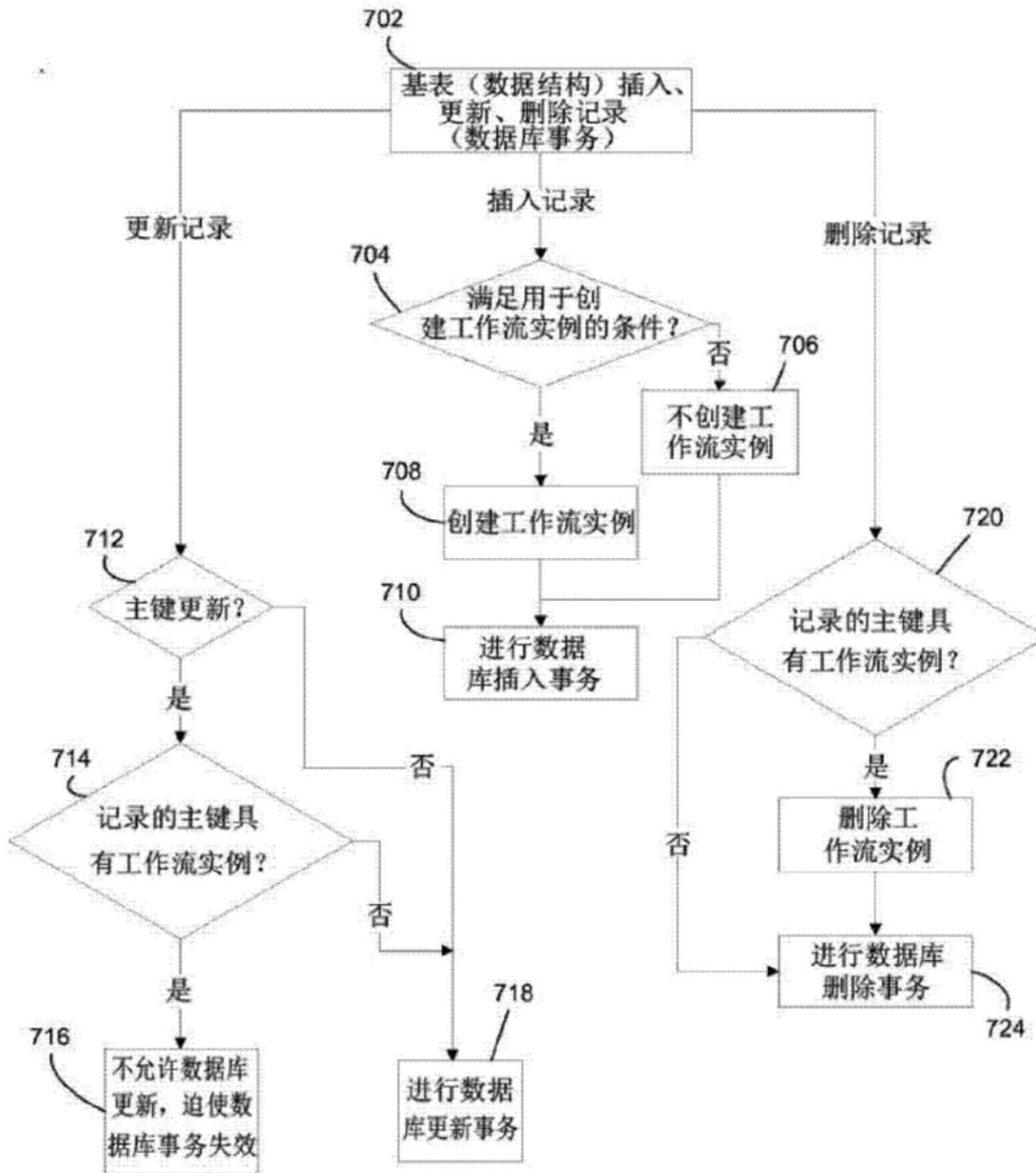
表504: workflow 事务对数据库表事务的作用	
workflow 事务	数据库基表事务
插入, 通过将现有的数据库记录恢复到 workflow 中	不动
更新, 当从一个状态移动到另一状态时	不动
删除, 当记录已到达结束状态时	不动

图5



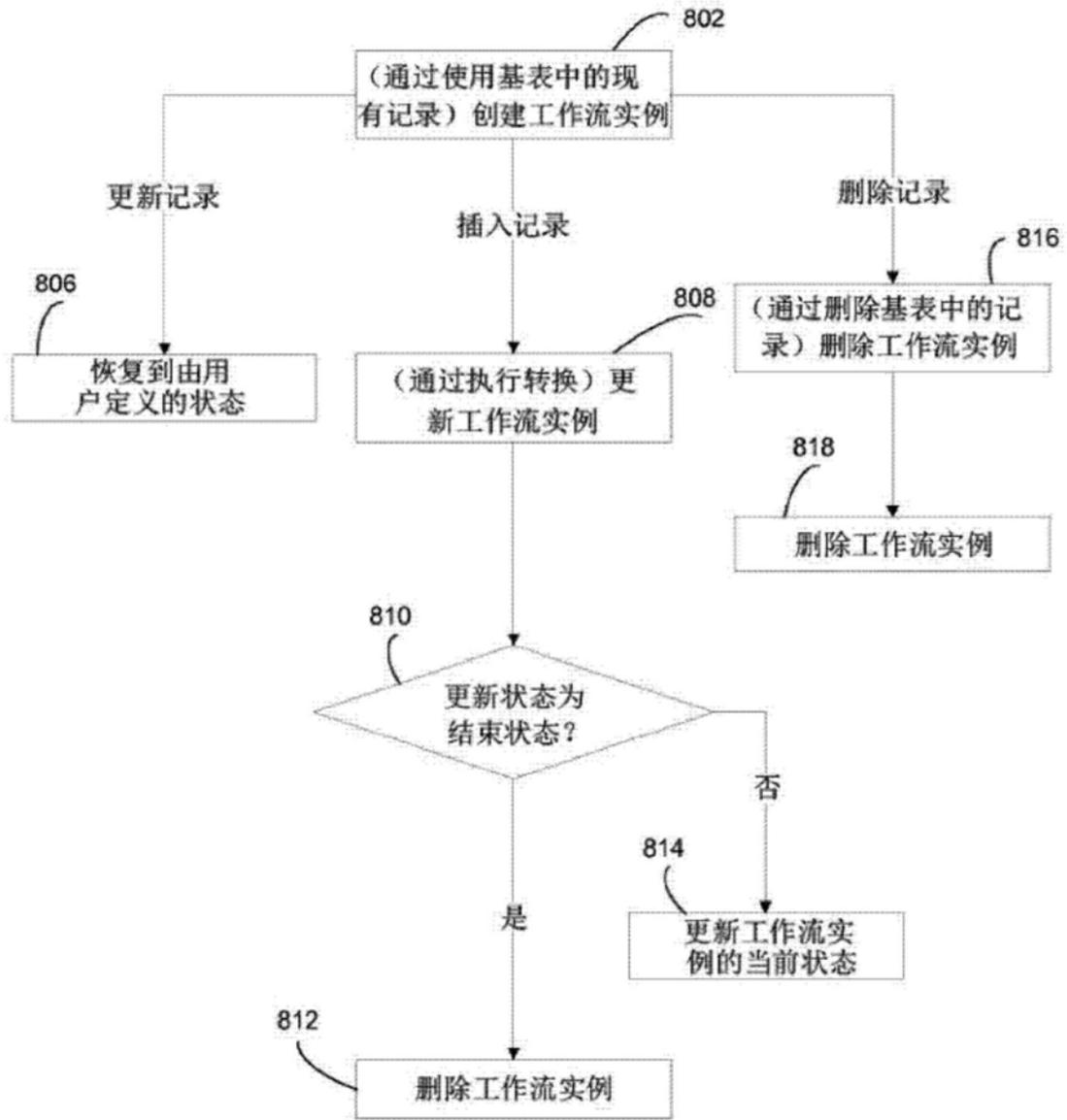
工作流启用和流程设计

图6



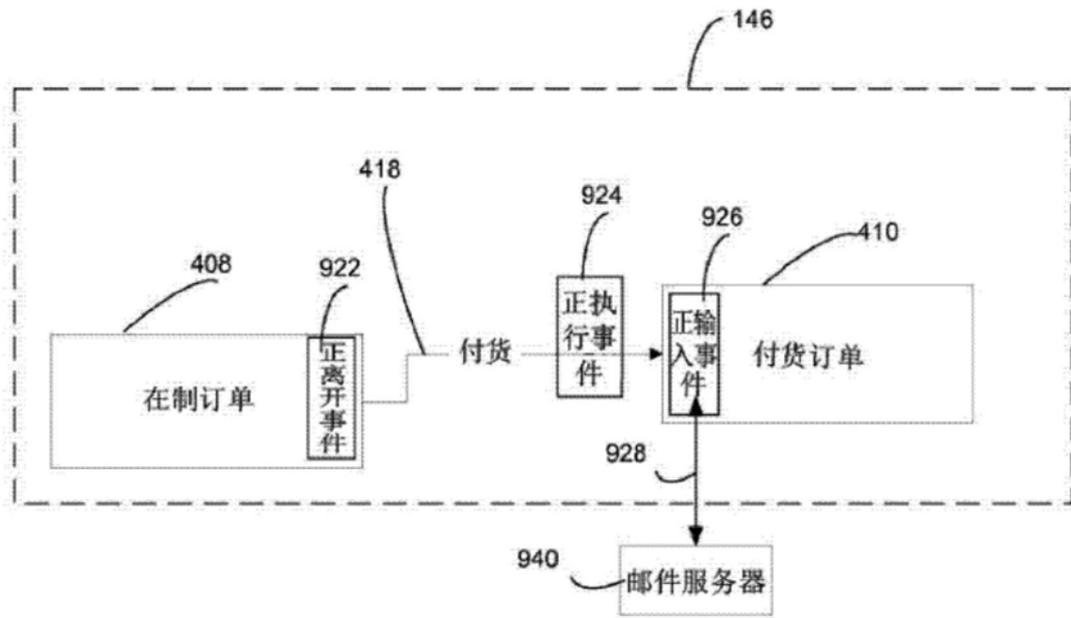
数据库 - 工作流事务

图7



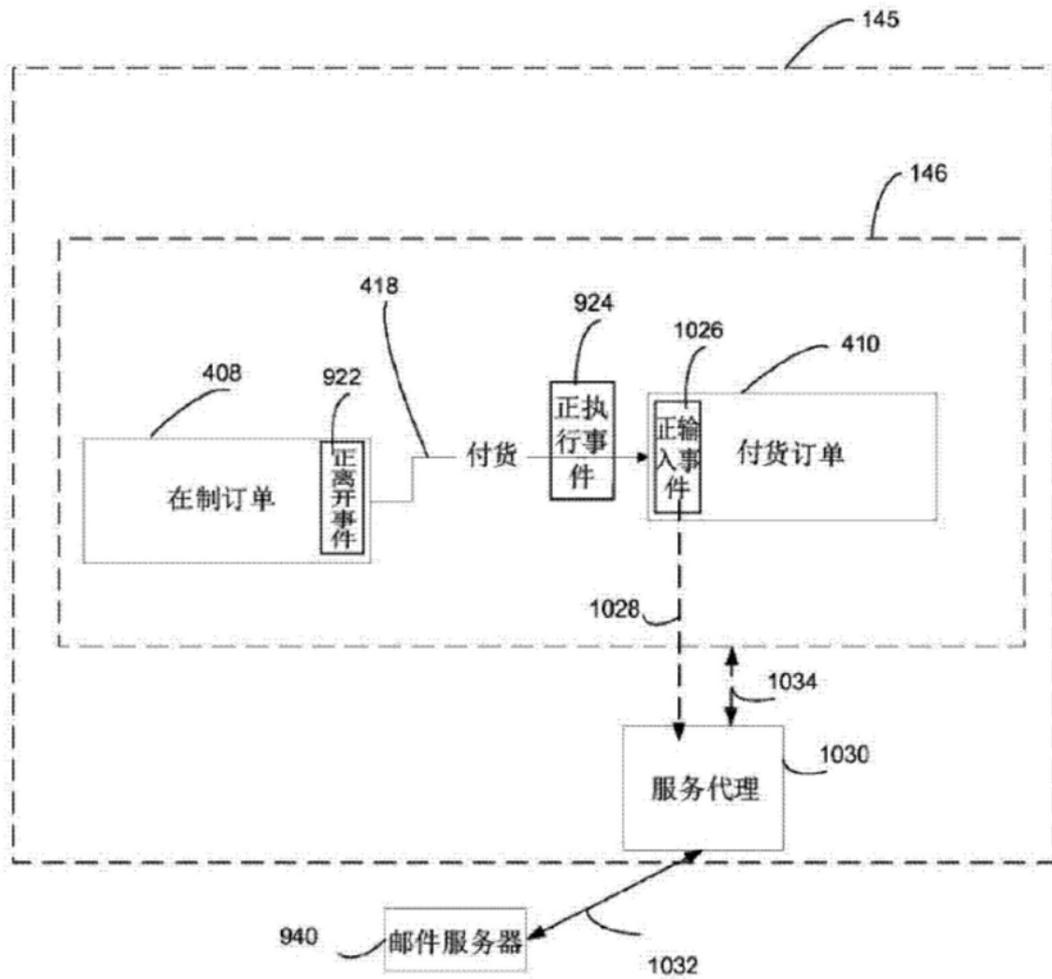
工作流事务

图8



workflow events

图9



使用服务代理的工作流事件

图10