



[12] 发明专利申请公开说明书

[21] 申请号 01814822.0

[43] 公开日 2005 年 8 月 24 日

[11] 公开号 CN 1659553A

[22] 申请日 2001.6.29 [21] 申请号 01814822.0
 [30] 优先权
 [32] 2000. 6. 29 [33] US [31] 60/214,910
 [32] 2001. 6. 29 [33] US [31] 09/896,559
 [86] 国际申请 PCT/US2001/020739 2001.6.29
 [87] 国际公布 WO2002/003294 英 2002.1.10
 [85] 进入国家阶段日期 2003.2.28
 [71] 申请人 美国联合包裹服务公司
 地址 美国佐治亚州
 [72] 发明人 辛迪·约翰逊 威尔·法斯蒂
 A·J·威尔森 蒂姆·扎克
 哈维·里可莱斯 肯·吉奥达诺
 凯文·麦克弗森 杰伊·鲍尔斯

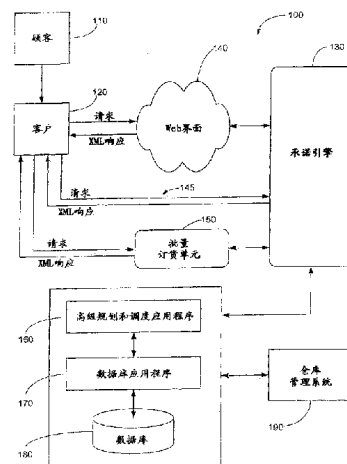
[74] 专利代理机构 中国国际贸易促进委员会专利商
 标事务所
 代理人 吴丽丽

权利要求书 2 页 说明书 24 页 附图 7 页

[54] 发明名称 用于端对端实现和供应链管理的系统和方法

[57] 摘要

一种供应链管理系统便于物品有效分配和发运。具有与其有关的地理位置的至少一个顾客与至少一个客户通信。该至少一个客户以顾客的名义生成一个订单，其中，该订单含有与至少一件物品相应的物品数据和与顾客地理位置相应的位置数据。在系统中，一个承诺引擎与该至少一个客户和与众多仓库进行通信。该承诺引擎部分根据与顾客地理位置相应的位置数据从众多仓库中标识出一个出货仓库，其中，出货仓库是离顾客地理位置最近的能在指定的时间内将至少一件物品发运给至少一个顾客的仓库。



1. 一种便于物品有效分配和发运的供应链管理系统, 包括:
至少一个顾客, 其中该至少一个顾客具有与其有关的顾客地理位置;
5 至少一个客户, 其中该至少一个客户与该至少一个顾客有关, 并且其中该至少一个客户生成一个订单, 该订单含有与至少一件物品相应的物品数据和与顾客地理位置相应的位置数据;
众多仓库, 其中众多仓库中的至少一个仓库与众多仓库中的至少另一个仓库分开, 并且其中众多仓库中的每个仓库都分别与顾客地理位置间隔一段地理距离; 和
10 一个与该至少一个客户和众多仓库通信的承诺引擎, 其中该承诺引擎部分根据与顾客地理位置相应的位置数据从众多仓库中标识出一个出货仓库,
其中, 该出货仓库是离顾客地理位置最近的能在指定的时间内将至少
15 一件物品发运给至少一个顾客的仓库。
2. 权利要求1的系统, 其中承诺引擎专门根据离顾客地理位置最近的能在指定的时间内将至少一件物品发运给至少一个顾客的仓库, 从众多仓库中选择一个出货仓库, 其中部分根据位置数据标识出这一离顾客地理位置最近的仓库。
- 20 3. 权利要求1的系统, 其中出货仓库是离顾客地理位置最近的在库存中具有该至少一件物品的仓库。
4. 权利要求1的系统, 其中所指定的时间由该至少一个顾客来确立。
5. 权利要求1的系统, 其中该至少一个客户提供具有至少一件物品的众多仓库的至少之一。
- 25 6. 权利要求1的系统, 其中客户通过广域网与承诺引擎通信。
7. 权利要求1的系统, 其中位置数据可以从含有邮政编码、国家、地区或者城市和州的集合中选择。
8. 权利要求1的系统, 还包括: 与用于将至少一件物品发运给至少一个顾客的承诺引擎有关的货运公司。
- 30 9. 一种便于物品有效分配和发运的方法, 包括:

生成至少一个订单，其中该至少一个订单含有与至少一件物品相应的物品数据和与顾客地理位置相应的位置数据；

评审该至少一个订单，以便部分根据与顾客地理位置相应的位置数据从众多仓库中标识出一个出货仓库；和

5 判断出货仓库是否有足够的库存，以便将该至少一件物品发运给顾客。

10. 权利要求9的方法，还包括：指令出货仓库将该至少一件物品发运给顾客。

10 11. 权利要求10的方法，其中出货仓库是离顾客地理位置最近的能在指定的时间内将至少一件物品发运给至少一个顾客的仓库。

12. 权利要求9的方法，其中根据与离顾客地理位置最近的仓库的关系选定出货仓库，并且该出货仓库能在指定的时间内将至少一件物品发运给至少一个顾客。

15 13. 权利要求9的方法，还包括：将至少一个订单从客户发送到承诺引擎，其中，承诺引擎评审该至少一个订单。

14. 权利要求13的方法，其中该至少一个订单从客户到承诺引擎的传送可以通过因特网来进行。

15. 权利要求9的方法，其中物品数据包括顾客运送条款。

16. 权利要求15的方法，其中顾客运送条款包括交货日期。

20 17. 权利要求9的方法，还包括：对顾客作出响应，以通知顾客该至少一件物品将被发运给该顾客。

18. 权利要求9的方法，其中出货仓库按接收到发运物品指令的次序来发运物品。

用于端对端实现和供应链管理的系统和方法

5 本申请享有2000年6月29日所申请的美国临时专利申请号60/214,910的优先权，该临时专利申请的名称为“METHODS, SYSTEMS AND COMPUTER PROGRAM PRODUCTS FOR REAL-TIME SUPPLY CHAIN MANAGEMENT”，其全部内容在此作为参考。

10 技术领域

本发明一般涉及供应链管理系统，尤其涉及以有成本收益的方式有效地向顾客分配和分发物品的订单实现和供应链管理系统。

背景技术

15 长期以来，人们已经知道，有效分发系统对在远程方之间进行交易的环境中运营的公司来说是十分重要的。这种远程交易的一个早期的普通例子是通过发出电话目录订单而在顾客与百货公司之间进行的远程交易。这种零售系统的便利和不间断的运营不仅依赖于顾客进行订购的能力而且依赖于零售商向顾客运送商品的能力。几十年来，分发系统一直执行这种
20 功能，尽管成功的程度有所不同。例如，大量的老式经济公司可以运营他们自己的分发系统，这些分发系统可以有效地处理大批订单，但不适应于不常发生的较小的出货。在该领域的另一个极端上，订单较少的小公司比如因特网零售商（或称电子零售商）可依赖于第三方分发系统，这是因为，他们没有分发方面的专长。然而，无论公司大小或所订购和发运给顾客的商品多少，能实现有效且有利的远程交易事务的一种共同特性是有效地
25 使订单从仓库到达顾客。

过去5年间，因特网使得远程交易飞速增长。今天，只要进行因特网连接，顾客就几乎能从世界上任何地方订购任何产品或服务。由于因特网使零售商能形成用来推销商品的电子商店，因此，加入其中的障碍极小，
30 从而导致了小型和中型公司广泛的存在于因特网上。小型和中型公司的因

特网存在的优点在于，公司可被大量顾客所访问，并且对顾客来说通过因特网访问这些公司就象访问传统的销售商品的真实商店那样容易。因此，因特网可为大大小小的公司都提供向用户展示和销售产品的手段。然而，任何公司无论大小都必须建立一种向在线顾客发运其产品的手段。尽管将

5 产品运送给顾客无论对普通的门市商店还是对电子零售商来说都是一个

问题，并且与商品从仓库传送给顾客有关的后勤发放几乎在所有将商品从远端发运给顾客的环境中都存在，然而，由于电子零售商缺乏因特网商业模式的经验并且由于在线买东西的顾客很多，因此后勤问题在电子商务领域尤其普遍。

10 为了保持竞争，小型和中型公司必须按时高效地交货，同时尽可能地减少成本。目前，这种后勤障碍对小型或中型公司的成功而言是最大的障碍之一。在因特网电子零售商的范畴，由于公司的发运机构不能发运顾客通过因特网web站点所订购的物品这一事实，因此常常不能按时交货。一些实体已经开始为电子零售商提供发运以及其他后端服务（比如，信用卡

15 处理），这样，电子零售商就能将焦点集中在其系统的前端。通过外购(out source)分发，减轻了公司与内部分发有关的额外开销费用（诸如配备职员、仓库设施维护以及分发费用）。

有许多第三方后勤提供商，他们为零售商提供第三方分发解决方案。这些提供商大多数都可提供从第三方后勤提供商已为此建立的一个仓库

20 发货。于是，第三方后勤提供商可通过与第三方后勤提供商有关的货运公司将电子零售商的产品从仓库发运给顾客。仓库分发提供商引用其自己的效率标准来减少物品处理时间，最大限度地减少与发运物品有关的费用，减小物品丢失或被偷窃的可能性，以及减小类似的仓库后勤隐患。然而，仓库通常难以控制与通过货运公司发运物品有关的费用。

25 因此，需要这样一种系统，该系统可以为公司提供入库和订单实现。具体地说，需要一种能快速、有成本收益地开展电子商务的端对端供应链管理业务。这种系统不仅要有利于小型至中型的因特网企业，而且要有利于逐渐重视因特网分发的老式传统企业。如果这种系统能在两天内从离顾客较近的众多仓库之一将产品发运到全国90%的地区，从而减少运载费

用,才具有优势。这种系统应根据顾客所在的位置以及仓库所具备的实现订单的能力来选择发运仓库。

发明内容

5 本发明的系统为客户(client)(包括厂商、批发商、零售商或分销商)预备可供使用的库存,顾客(customer)可针对这些客户查询和/或订购物品。客户可以以多种方式(比如,通过客户所建立的web网页,利用通过专用网等发送的标准EDI格式,或利用客户与其顾客预先安排的某种其他媒体)来接收顾客的查询。一旦客户与其顾客进行交互以确定顾客的需求,
10 客户就向承诺引擎发送一个承诺请求或订单请求,以判断顾客所请求的物品是否在库存中以及能否在所要求的日期之前投送到所要求的地点。承诺引擎负责接收和响应顾客请求,为客户预备可供使用的库存,并指令一个或多个分发中心(这里是称为仓库)实现从客户接收到的订单。根据本发明一个方面,承诺引擎部分根据与顾客地理位置相应的位置数据从众
15 多仓库中标识出一个出货仓库,其中该出货仓库是离顾客地理位置最近的能在指定的时间内将至少一件物品发运给至少一个顾客的仓库。因此,该系统使得能预留物品和/或从最适合于将这些物品发运给顾客的一个或多个仓库中发运出这些物品。

根据本发明一种实施方式,公开了一种便于商品(这里也称为物品)
20 有效分配和发运的供应链管理系统。该供应链管理系统包括:至少一个顾客,其中该至少一个顾客具有与其有关的顾客地理位置;和至少一个客户,其中该至少一个客户与该至少一个顾客有关,并且其中该至少一个客户生成一个订单,该订单含有与至少一件物品相应的物品数据和与顾客地理位置相应的位置数据。该系统还包括众多仓库,其中众多仓库中的至少一个
25 仓库与众多仓库中的至少另一个仓库分开,并且其中众多仓库中的每个仓库都与分别与顾客地理位置间隔一段地理距离。系统中还包括一个与该至少一个客户和众多仓库通信的承诺引擎,并且该承诺引擎部分根据与顾客地理位置相应的位置数据从众多仓库中标识出一个出货仓库。此外,在该供应链管理系统中,出货仓库是离顾客地理位置最近的能在指定的时间内
30 将至少一件物品发运给至少一个顾客的仓库。

根据本发明的一个方面，承诺引擎专门根据离顾客地理位置最近的能在指定的时间内将至少一件物品发运给至少一个顾客的仓库，从众多仓库中选择一个出货仓库，其中部分根据位置数据标识出这一离顾客地理位置最近的仓库。根据本发明的另一方面，所指定的时间由该至少一个顾客来确立。根据本发明的又一方面，出货仓库是离顾客地理位置最近的在其库存中具有该至少一件物品的仓库。位置数据可以从邮政编码、国家、地区或者城市和州中选择。通常，一个或多个货运公司与用于将至少一件物品发运给至少一个顾客的承诺引擎有关。根据本发明的另一方面，至少一个客户向众多仓库的至少之一提供至少一件物品。另外，客户可能通过广域网与承诺引擎通信。

根据本发明的另一实施方式，公开了一种便于商品有效分配和发运的方法。该方法包括：生成至少一个订单，其中该至少一个订单含有与至少一件物品相应的物品数据和与顾客地理位置相应的位置数据；评审该至少一个订单，以便部分根据与顾客地理位置相应的位置数据从众多仓库中标识出一个出货仓库；和判断出货仓库是否有足够的库存，以便将至少一件物品发运给顾客。

根据本发明的一个方面，该方法还包括：指令出货仓库将该至少一件物品发运给顾客。出货仓库可以是离顾客地理位置最近的能在指定的时间内将至少一件物品发运给至少一个顾客的仓库。或者，可以根据与离顾客地理位置最近的仓库的关系选定出货仓库，其中该出货仓库能在指定的时间内将至少一件物品发运给至少一个顾客。

根据本发明的另一方面，该方法还包括如下步骤：将至少一个订单从客户发送到承诺引擎，其中，承诺引擎评审该至少一个订单。该至少一个订单从客户到承诺引擎的传送还可以通过因特网来进行。根据本发明的另一方面，物品数据可以包括顾客运送条款，比如交货日期或所要求的交货方式。在顾客提交了至少一个订单后，该方法可以包括对顾客进行响应，以通知顾客该至少一件物品将被发运给该顾客。最后，出货仓库可以以先进先出的方式进行操作，这样，出货仓库将按接收到发运物品指令的次序来发运物品。

对熟练技术人员而言,根据下列附图以及详细描述将可以看到本发明的其他特征和优点。在附属权利要求所规定的本发明的范围内,已包含了所有这些特征和优点。

5 附图说明

因此,在概括描述了本发明后,下面将参照附图(这些图未必是按比例绘制的)进行详述,其中:

图1根据本发明的一种实施方式示出了本发明的系统的框图。

图2根据本发明的一种实施方式示出了承诺引擎的框图。

10 图3根据本发明的一个方面示出了在客户与承诺引擎之间的通信流程。

图4根据本发明的另一方面示出了在客户与承诺引擎之间的通信流程。

15 图5A是根据本发明的一种优选实施方式的流程图,该图说明了承诺引擎判断分发系统中哪个仓库将试图满足请求时所采用的规则。

图5B是根据本发明的另一种实施方式的流程图,该图说明了承诺引擎判断分发系统中哪个仓库将试图满足请求时可以采用的另一组规则。

图5C根据所举一个例子示出了与向本发明的系统请求物品的两个顾客的位置相应的四个仓库的位置。

20

具体实施方式

下面,将参照这些说明了本发明的优选实施方式的附图来详述本发明。不过,本发明可以以许多不同的方式来体现,而并不应当局限于这里所陈述的实施方式;相反,对熟练技术人员而言,通过提供这些实施方式,25 所揭示的内容将是充分的和完整的,并且完全能说明本发明的范围。全文中,相似的标号表示相似的单元。

正如熟练技术人员所知,本发明可以体现为一种方法、一种数据处理系统或一种计算机程序产品。因此,本发明可以采取以下形式:完全硬件的实施方式,完全软件的实施方式,或兼有软件和硬件两方面的实施方式。30 再者,本发明还可以采取以下形式:计算机可读存储媒体上的计算机程序

产品，该存储媒体中包含有计算机可读程序代码装置。具体地说，本发明可以采取web实现的计算机软件的形式。任何合适的计算机可读存储媒体都可以使用，包括硬盘、CD-ROM、光存储器件或磁存储器件。

下面，参照根据本发明的一种实施方式的系统、方法、设备和计算机程序产品的框图和流程图图解来描述本发明。应当理解，这些框图和流程图图解中的每个块以及框图和流程图图解中的块的组合分别都可以用计算机程序指令来实现。这些计算机程序指令可以装载在通用计算机、专用计算机或者其他可编程数据处理设备上，以产生机器代码，这样，在计算机或其他可编程数据处理设备上所执行的这些指令将形成用于实现这一流程块或这些流程块中所指定功能的工具。

这些计算机程序指令还可以存储在计算机可读存储器中，它能指令计算机或其他可编程数据处理设备以特定方式运行，这样，计算机可读存储器中所存储的指令就形成了含有实现这一流程块或这些流程块中所指定功能的指令装置的一种产品。计算机程序指令还可以装载在计算机或其他可编程数据处理设备上，以形成该计算机或其他可编程设备上所要执行的一系列操作步骤，从而产生计算机实现的过程，这样，计算机或其他可编程设备上所执行的这些指令将提供用于实现这一流程块或这些流程块中所指定功能的步骤。

相应地，框图和流程图图解的块支持用于实现指定功能的装置的组合、用于实现指定功能的步骤的组合以及用于实现指定功能的程序指令装置。还应当理解，这些框图和流程图图解中的每个块以及框图和流程图图解中的块的组合可以由能执行指定功能或步骤的基于专用硬件的计算机系统来实现，或者由专用硬件和计算机指令的组合来实现。

下面，参照图1，本发明的供应链管理系统100便于接收来自顾客110的关于与承诺引擎130相关的货运公司（未示出）发运给顾客110的物品的查询和订单（这里也统称为请求）。顾客与客户120交互，以便发出物品订单或查询，然后，客户120又以顾客的名义将一些表示顾客查询和/或订单的请求转发给承诺引擎120。承诺引擎130负责接收和响应顾客110的请求，包括为客户预备可供使用的库存，并负责指令一个或多个仓库实现从客户

120接收到的订单。因此，系统100使得能预留物品和/或从最适合于将这些物品发运给顾客110的一个或多个仓库中发运出这些物品。

客户120可以以多种方式（比如，通过客户所建立的web网页，利用通过专用网等发送的标准EDI格式，或利用客户与其顾客预先安排的某种其他媒体）来接收顾客的查询。一旦客户120与其顾客110进行交互以确定顾客的需求，客户120就向承诺引擎130发送一个承诺请求或订单请求，以判断顾客所请求的物品是否在库存中以及能否在所要求的日期之前投送到所要求的地点。下面将详细讨论承诺请求和订单请求。如图1中所示，客户120直接通过web界面140和/或通过批量订货单元（batch order singulator）150与承诺引擎130通信。

web界面140最好包括至少一个客户web网页，顾客110或客户120可以通过该网页请求将物品发运给顾客110，或者可以请求承诺引擎130在一段临时时间段内预留或分配物品，如以下所详述。例如，在因特网零售商环境中，web界面140可以包括因特网零售商所建立的因特网web网页，零售顾客通过该网页请求发运一件或多件所订购的物品。正如普通技术人员所知，web界面140还可以包括服务器，用于建立和启动与至少一个客户web网页的通信。尽管web界面140最好以扩展标记语言（XML）格式接收来自客户的通信，然而web界面140也可以以任何适合于转发标识查询和订单的必要数据的格式接收来自客户120的查询和订单。不过，在系统100的优选实施方式中采用了XML，这是因为，XML尤其有利于发送非结构化数据记录（这种非结构化数据记录是系统100中的数据记录的优选格式）。应当注意，根据本发明的一个方面，数据以XML格式被承诺引擎110所接收和发送，因此，承诺引擎110的响应也是XML格式。这种数据格式可以以复合数据记录的形式将数据转发到web界面140，这种复合数据记录包括了标识众多查询和/或订单的数据，还包括诸如顾客身份、运达地点信息、所请求的交货日期等信息。这一数据可以与一个或多个订单有关，其中每个订单都占一行或多行，比如在一个数据记录中。

客户请求可以包括通过web界面120实时接收到的或通过批量130转发所接收到的请求。此外，这些请求可以包括具有逗号界定值的电子数据交换（EDI）格式在内的多种格式被接收。一旦web界面140接收到这些请

求, 界面140就将这些请求转换成XML文档(无论接收订单采用什么格式)。将请求转换成XML文档或XML格式化的请求对普通技术人员而言是众所周知的, 因此在这里不作详细讨论。由于这种转换, 无论客户120提供什么样的请求格式, 承诺引擎110都只接收XML格式化的请求。

5 如图1中所示, 客户120还可以通过直接链路145或通过批量订货单元150与承诺引擎130通信。在客户通过直接链路145与承诺引擎130通信的情况下, 客户120可以通过广域网(WAN)、局域网(LAN)、专用线、普通拨号连接或普通技术人员熟悉的类似方法与承诺引擎130通信。根据本发明的一个方面, 从客户到承诺引擎130的直接通信最好采用XML格式, 这种
10 格式对发送复杂数据记录来说是一种方便的格式。同样, 从承诺引擎130通过直接链路145到客户120的通信最好也采用XML格式(如图1中示为XML响应)。这就免去了客户和/或承诺引擎要包含转换软件的需要, 这种转换软件对通过web界面140或通过批量订货单元150通信的情况而言是不需要的。不过, 应当理解, 也可以采用熟练技术人员熟悉的其他格式通过直
15 接链路145来发送数据。

通过直接链路145发送到承诺引擎130的数据其内容与前面所述的实施方式中通过web界面140发送到承诺引擎130的数据类似。然而, 正如熟练技术人员所知, 不同于从web界面140传送到承诺引擎130的通信(它采用安全超文本传输协议), 通过直接链路145的通信采用文件传输协议
20 (FTP)的形式。在客户与承诺引擎130之间采用直接链路145将不需要依赖于因特网通信, 这种因特网通信需要因特网业务提供商(ISP)并且可能不如直接通信可靠。因此, 在因特网、WAN等连接不可用时, 直接链路145是可取的。然而, 直接链路145不可能提供方便的图形用户界面或始终贯通的连接性。此外, 在承诺引擎130要求通信采用指定格式比如XML的情况下, 客户120或承诺引擎130必须将任何这样的通信都转换成XML格式,
25 这种功能在前面的实施方式中也可以由web界面140来提供。

客户120可以与承诺引擎130通信(反之亦然)所采用的第三种方法是通过批量订货单元150, 该批量订货单元可以位于客户120与承诺引擎130之间的通信通路上中。因此, 批量订货单元150可以位于LAN、WAN等之中,
30 并且也可以位于客户120和承诺引擎130两者之一中或这两者之中。批量订

货单元150的用途在于标识和/或分离以批量方式、从同时向承诺引擎130提交了用于发运的若干件物品的客户接收到的请求。因此，在客户120向承诺引擎130提交众多请求的情况下，最好通过批量订货单元将这些请求发送到承诺引擎130。批量订货单元150还可以将响应不加修改地从承诺引擎转发到客户120，或者，可以收集从承诺引擎130接收到的多个XML响应并将这些响应合成为转发给客户120的一个通信。普通技术人员应当理解，批量订货单元150的功能可以由客户120也可以由承诺引擎130来实现。此外，针对通过因特网发送到承诺引擎130的请求，web界面140可以实现与批量订货单元类似的功能。再者，通过批量订货单元从客户120发送到承诺引擎130的通信（比如请求）也可能导致通过上述其他方法之一（比如通过直接链路145或通过web界面140）从承诺引擎130发送到客户120的响应。

如上所述，承诺引擎130负责接收和响应顾客请求，包括为客户预备可供使用的库存，并负责指令一个或多个仓库实现从客户120接收到的请求。具体地说，承诺引擎130接收承诺请求和订单请求，并判断顾客所请求的物品是否在库存中以及能否在所要求的日期之前投送到所要求的地点。承诺请求从承诺引擎130中查找到一个承诺，即物品可以得到并能发运给顾客，以满足顾客和/或客户运送条款。承诺请求并不约束客户。承诺引擎130将该物品预留一段预定时间；然而，承诺引擎将不指令出货仓库将物品发运给顾客。在这段预留时间过后，承诺引擎130通知出货仓库该物品不应再作预留标记。

订单请求虽与承诺请求类似，但它是这样一种请求，即可能的话承诺引擎130将实现订单。承诺引擎130作出的能实现所请求订单的响应将责令承诺引擎130提供该订单，除非顾客或客户在启动该订单所标识的物品出货之前改变了请求。因此，如果承诺引擎130能满足顾客和/或客户运送条款，那么承诺引擎130将指令出货仓库实现订单请求。

承诺请求和订单请求具有相同的普通格式，并且含有相同的数据，但订单请求可能含有一个标识符以使订单请求与以前所提供的承诺相符，这样，承诺请求之后的订单请求就不必再提交数据，因为，在订单请求中的运送条款和物品与以前所发送的承诺请求中的内容是完全相同的。承诺引

擎130可以通过标识了请求类型的数据（比如与每个请求有关的标志）识别出请求的种类（承诺或订单）。一个请求的细节包括物品、物品数量、顾客位置数据、发运选项、交货日期、顾客邮政编码、地址、客户唯一ID码等通常与订单有关的项目。一个请求可以包括多种物品，每种物品都有
5 交货日期和相关的运达位置。顾客所输入的地理位置是需要的，这是因为，如有订货，系统（具体地说是承诺引擎130）将用到这一地理位置，以确定发运所请求的物品的仓库。

除了与物品相应的物品数据和位置数据之外，该请求还可能包括标识所需增值业务（VAS）的物品数据。VAS物品的一个例子是礼品包装纸，它
10 要求随所请求的实际物品一起发送到承诺引擎130。在承诺引擎识别出这一数据的情况下，承诺引擎130将随实际物品一起检查礼品包装纸的仓库库存和容量（capacity）。应当理解，在订单中可以有若干件物品，其中只有一个子集要求礼品包装。在这种情况下，只有要进行礼品包装的物品才包含在具有礼品包装纸的出货集合中（即要发运的产品的组中），这样，
15 承诺引擎130在选择仓库来实现请求之前需要两个判据都满足。此外，还可以有其他VAS物品，比如插页和目录册，这些物品将包括在订单中，但没有相关的库存或容量核对。没有足够的储备或容量来包括这几种业务并不会阻止订单出货。这些业务需要出现在订单上，从而一旦订单付诸实施（仓库），就会包括它们。

20 下面再来参照图1中的系统100，承诺引擎130被执行时将与高级规划和调度（APS）应用程序160相结合，在订单实现系统100中，这一应用程序提供了库存和订单管理、应收帐单跟踪以及高级规划和调度。尽管承诺引擎130可以纳入APS应用程序160的特性，从而可以在没有单独的APS应用程序160的情况下被实现，然而，本发明的优选实施方式使用了常规APS
25 应用程序160。根据本发明的一个方面，APS应用程序160包括一个Oracle Iii的APS应用程序，这种程序提供了系统100所必要的一些功能。这些功能包括全球承诺可用性（global available to promise）（ATP）和高级供应链计划，这些功能在技术上都是众所周知的。APS应用程序160还可以支持一些附加功能，比如，需求计划和生产调度，但这些附加功能对本发明
30 的优选实施方式而言无关紧要。

APS应用程序160的全球ATP (GATP) 组件可以包括一个GATP应用程序或服务器, 它管理数据库调用, 以说明在特定时刻系统库存的状态, 从而保持特定时刻系统100承诺可用的所有库存的瞬态图。因此, 承诺引擎130查询GAPS以判断仓库库存的状态。数据(比如储备的库存)的更新由数据库应用程序170提供给APS应用程序160, 该数据库应用程序最好是另一个现成的常规组件, 比如, Oracle 8I数据库应用程序。此外, 也可以采用一个或多个核心模块, 比如库存和订单管理应用程序。数据库应用程序170更新位于仓库层中的一个或多个数据库180中的GATP库存。每个仓库都包括一个仓库管理系统190, 该仓库管理系统使得可以进行订单的接收、实现和确认, 以及与该仓库有关的一个或多个数据库190的更新。利用数据库应用程序170的性能, APS应用程序的GATP功能(或应用程序)通过仓库管理系统190所提供的出货认可保留一个具有打开的(即未实现的)订单和关闭的(即已出货的)订单的画面。总之, APS应用程序160只是起到了用于供需信息的储存库的作用。

因此, 承诺引擎 130 是系统 100 的智能部分, 并查询 GATP 以判断系统 100 是否有在特定日期要从特定仓库出货的特定物品。承诺引擎 130 用于查询每个仓库的库存的订单将在下面进行详述, 它与顾客的地理位置有关。GATP 在将应答发回给承诺引擎 130 后, 承诺引擎 130 将检查这一应答, 以判断订单能否由某个仓库来实现。如果不能, 那么, 承诺引擎判断另一个仓库能否满足这一请求。然后, 承诺引擎 130 以 XML 响应对顾客请求作出响应, 如图 3 中所示。如果仓库能满足订单, 并且该请求是一个订单请求, 那么, 承诺引擎 130 通过 APS 应用程序 160 将指令转发给仓库管理系统 190, 以便发运所订购物品。

图2根据本发明的一种实施方式示出了图1的系统中所述的承诺引擎230的框图。承诺引擎230包括处理器260、存储设备280、存储器240、输入输出设备270和网络接口290。承诺引擎处理器260内的各个单元都通过总线210互相通信。此外, 网络接口290使得承诺引擎230可以利用已知硬件通过WAN、LAN、PSTN、分组交换网、因特网等与计算机网络的其他单元连接并进行接口和通信。

存储器240包括操作系统250和承诺引擎控制模块220。承诺引擎控

制模块 220 包括用于控制承诺引擎操作的软件，这将在以下参照图 3-5 进行描述。承诺引擎控制模块 220 在操作系统 250 和处理器 260 的协助下进行操作。操作系统可以是众所周知的任何操作系统，如 Windows NT，它执行承诺引擎控制模块 220 的指令，并便于承诺引擎控制模块 220 与其他
5 承诺引擎 230 单元进行通信，或与那些在承诺引擎 230 外部并通过网络接口 290 与承诺引擎 230 通信的网络单元进行通信。

存储设备 280 代表至少一个存储设备，比如，硬盘驱动器、软盘驱动器、CD-ROM 驱动器或光盘驱动器，用于在各种计算机可读媒体（如硬盘、可换磁盘、CD-ROM 等）上存储信息。存储设备 280 及其有关的计算机可
10 读媒体为承诺引擎提供了非易失性存储。重要的是应注意，该计算机可读媒体可用本领域中已知的其他任何类型的计算机可读媒体来取代。这种媒体可以包括例如盒式磁带、闪速存储器卡、数字化视频光盘和 Bernoulli 盒式磁带。

普通技术人员应当了解，一个或多个承诺引擎 230 组件其地理位置可以
15 远离其他的承诺引擎 230 组件。此外，一个或多个这种组件也可以结合在一起，并且执行这里所述的功能的其他组件可以包含在承诺引擎 230 中。

下面参照图 3，当客户 320 向承诺引擎 330 发出一个承诺请求（RFP）
340 时，对特定交货日期的物品作出承诺的过程便开始。承诺请求 340 包
20 括所需的用于评估所请求物品的可用性的信息。例如，该请求包括与一件或多件所请求物品相应的物品数据、与顾客的地理位置相应的位置数据、运送条款以及处理该请求所需的其他数据。如果物品在所需的考虑到了出货选项的时间内可以发运给客户的顾客，那么，承诺引擎 330 向客户 320 返回一个承诺 350。这一承诺 350 是物品的临时预备。如果客户想锁定（遵
25 守）请求的条款和相关承诺，那么，客户将向承诺引擎 330 发送一个订单请求（RFO）360。订单请求 360 是客户想遵守请求的条款的请求。订单请求 360 还要有标识符，以使得承诺引擎可以使该订单请求 360 与以前提供的承诺 350 结合。这一标识符将使承诺引擎 330 可以识别在以前提供的承诺请求中客户所请求的物品，而无需再将这种信息重发给承诺引擎 330。
30 当请求 360 与承诺 350 相符时，承诺引擎 330 向客户发出一个认可 370，

以便完成交易并约束客户和承诺引擎 330 都遵照订单请求 360 的条款。如果没有发送订单请求，那么，承诺将期满并且承诺引擎 330 将不受约束。

如图 4 中所示，客户 400 可以在事先没有提交承诺请求的情况下向承诺引擎 430 发送订单请求 460。这样，客户 420 将遵守订单请求 460 的条款，如果承诺引擎 430 可以接受的话。当然，由于没有可匹配的事先承诺，因此，承诺引擎 430 是通过判断物品的可用性开始的。如果在订单请求 460 中请求的物品是可获得的，那么承诺引擎 430 可以向客户 420 发出一个认可 470，这一认可可以约束客户和承诺引擎 430 都遵照订单请求 420 的条款，即形成一个订单。

图 4 中所示的流程与在相关的先前提提供的承诺（如图 3 中所示）已期满时处理订单请求的过程相似。在这种情况下，将处理订单请求 460，就好像没有发送以前的承诺请求。如果物品是可获得的，那么，将形成一个订单，并向客户发回一个认可 470。此外，如果订单请求以先前提提供的承诺的标识符发给承诺引擎 430 但却具有与原先的承诺请求不同的条款，那么，承诺引擎 430 将根据图 4 判断可用性，就好像以前的承诺不存在。然而，承诺引擎 430 仍可以用该标识符来标识交易，而无需与订单请求一同向承诺引擎 430 重新发送交易的标识。因此，图 4 中所示的处理流程可能与图 3 中的最后两个步骤中的流程不同，这是因为，在图 3 的流程中，标识符包含在与已流逝的承诺相应的订单请求中。

接着，图 5A 根据本发明的优选实施方式示出了承诺引擎判断该分发系统中哪个仓库将试图满足请求时所采用的基本规则。在接收到来自客户的请求后，承诺引擎 130 首先对请求进行检查，以确定离顾客最近的仓库（块 500）。为了确定最靠近的仓库，承诺引擎 130（具体地说，承诺控制模块 220）查询承诺引擎中所保存的交货网络数据，以确定地理上与请求中的顾客地理位置数据最接近的仓库。因此，交货网络数据包括每个仓库的位置信息，这些位置信息可以存储在承诺引擎的存储设备 280 中或者承诺引擎控制模块 220 中。承诺引擎 130 将离顾客地理位置最近的仓库指定为请求所起源的首要仓库，这是因为，它离顾客位置最近，从而认为它可以以最低的费用将物品发运给顾客。一旦确定了首要仓库，承诺引擎 130 就查阅（或查找）与该首要仓库有关的次要仓库的预定顺序列表（块 505）。

次要仓库在首要仓库之后以同一方法按优先次序排序，而与请求中的运达地理位置无关。次要仓库将根据离首要仓库的距离来区分先后次序，这意味着，根据次要仓库相对于首要仓库的位置，次要仓库未必总是离各运达位置最近。

- 5 在确定了该仓库表后，承诺引擎控制模块 220 判断最近的（首要）仓库能否满足请求（块 510）。为了作出这一判断，承诺引擎 130 与 APS 应用程序 160 进行通信，该应用程序保存了系统中每个仓库的最新库存。具体地说，承诺引擎将查询 APS 应用程序，试图查明首要仓库能否满足请求中所标识的判据，比如仓库是否包括所请求的这一或这些物品，能否满足
- 10 顾客或客户指定的运送条款，以及能否满足该请求的任何附加要求。APS 以与承诺引擎的查询相应的数据作出响应，而承诺引擎判断首要仓库能否满足这一请求。若能，则选定首要仓库来满足请求（块 550）。

- 如果首要仓库不能满足这一请求，那么，承诺引擎 130 判断是否存在与该首要仓库有关的别的仓库（块 515）。通常，系统中的所有仓库都与
- 15 相应表（其中每个仓库是首要仓库）中的其他仓库有关。这些仓库按离首要仓库最近至最远的次序排列。可选地，承诺引擎 130 可以只在判断了最近的仓库能否满足请求后，查找与首要仓库有关的仓库表，这样，在图 5A 的流程图中，块 505 位于块 510 与 515 之间。如果这些仓库中任何一个都不能满足这一请求，那么，承诺引擎用一个与顾客和/或客户的通信
- 20 来表示这种情况（块 540）。然而，必要的话，承诺引擎 130 可以先尝试在两个或多个仓库之间分别满足该订单，以满足任一仓库都不能满足的请求。这将在以后作更详细的描述。否则，承诺引擎 130 重复以上所讨论的步骤，并判断下一个最近的仓库能否满足这一请求（块 520、525、530）。这一过程一直顺延进行下去，直到选定一个仓库（块 550）或者没有仓库
- 25 能满足这一请求（块 540）为止。应当理解，尽管图 5A 的流程图表示承诺引擎顺延地判断各仓库能否满足这一请求，从而减轻确定满足请求的合适仓库所需的处理过程，承诺引擎可以同时请求所有仓库的这种信息。在这种情况下，只可以向 APS 应用程序提交一个请求。不过，承诺引擎将以与以上所讨论相同的方式处理 APS 应用程序响应。

- 30 图 5B 根据本发明的另一实施方式示出了承诺引擎判断该分发系统中

的哪个仓库将试图满足请求时可以采用的另一组规则。这一实施方式与图 5A 的实施方式的不同之处在于，承诺引擎始终试图利用离顾客最近的仓库来满足各请求。在确定了离顾客最近的（首要）仓库（块 552）之后，正如图 5A 的块 500 中那样，承诺引擎判断该首要仓库能否满足这一请求（块 555），正如图 5A 的块 510 中那样。若能，则选定该首要仓库（块 580）。否则，承诺引擎确定能满足这一请求的下一个离顾客最近的仓库。这一点与图 5A 中所示的过程不同，因为承诺引擎是确定下一个离顾客最近的仓库而不是查阅保存有仓库的相互间距表的交货网络数据。只要有可以查询的别的仓库，这一过程一直进行（块 560、565、570），直到不能满足这一请求（块 575）或者选定一个仓库（块 580）为止。

图 5C 是表示与向分发系统 500 请求物品的两个顾客的位置相应的四个仓库的位置的框图。该框图包括首要仓库（“A”）586、第二仓库（“B”）590、第三仓库（“B”）588 和第四仓库（“D”）592。在框图中，A 586 离 C 590 的距离比离 B 588 的距离更近，以表示它在地理上离 C 比离 B 更近。还示出了在第一地理位置的第一顾客 596 和在第二地理位置的第二顾客 594。接下来，图 5C 用来说明以上图 5A 和 5B 的两种可供选择的过程。

采用图 5A 的过程时，来自第一顾客 596 和第二顾客 594 的请求被提交给与这两个顾客 596、594 有关的客户的承诺引擎。这些请求使处理引擎作出这样的决定：可能的话，应当用 A 586 来满足这两个请求，因为它在地理位置上离第一顾客 596 和第二顾客 594 最近。参照下面假设的表 1，如果 A 586 被认为是首要仓库，那么次要仓库的次序将是 C 590，然后是 B 588，再然后是 D 592，而与第一顾客的位置无关。这基于这样的事实：C 590 离 A 586 的距离比 B 588 或 D 592 离 A 586 的距离更近，而 D 592 离 A 586 最远。因此，尽管仓库 C590 可能是第一请求的下一最近仓库，但它不是第二请求的下一最近仓库，这是因为，第二请求实际上离 B 588 的距离比离 C 590 的距离更近。因此，如果 A 586 不能满足这些请求而 C 590 和 B 588 都能满足这些请求，那么，承诺引擎（具体地说是承诺引擎控制模块）将指令 C 590 来满足这两个请求，因为，根据表 1 中所示的交货网络数据，C 590 是第二仓库。本例中，有四种不同的寻源规则，其中每种规则与系统中所规定的每个仓库是一一对应的。系统中的仓库有多少

个，寻源规则就有多少种。

首要仓库	A	B	C	D
第二仓库	C	A	A	A
第三仓库	B	D	D	B
第四仓库	D	C	B	C

表 1

- 5 因此，采用图5A中的过程时，承诺引擎首先将试图从最靠近的可供使用仓库来满足请求，然后将利用预定的地理关系来确定仓库被要求满足请求的次序。这一过程最大限度地减少了处理引擎所需要的处理过程，从而更优于图5B的过程。

- 10 图5B的可选过程会导致不同的结果。由于图5B中的过程总是要找出离顾客最近的仓库，因此，如果A 586不能满足两个请求（一个来自第一顾客，一个来自第二顾客），那么，第一顾客请求将由C 590来满足而第二顾客请求将由B 588（而不是上例中的C）来满足，只要B 588和C 590能满足这些请求。于是，承诺引擎操作的前提是：离顾客最近的可供使用仓库具有最低的发运费用，并且总要选择下一最近仓库，而不管导致这一
- 15 结果所需的处理。

A. 请求类型

- 至此，对承诺引擎130的请求被描述成承诺请求和订单请求。具体地说，客户针对承诺或订单可以作出四种请求：无限制，有日期限制，有服务限制，和加急（这是有服务限制的特例）。这些订单由与一件或多件所
- 20 请求物品相应的物品数据、运送条款或者另外的处理请求所需要的数据来标识。针对所有四种请求类型，以上所讨论的承诺规则由承诺引擎控制模块220来执行。然而，针对每种单一的请求类型，还有一些独特的承诺流程规则。这些规则同样由承诺引擎控制模块220来维持。

- 通常，无限制、有日期限制和有服务限制是标准类型的订单或请求。
- 25 这些请求或订单利用仓库的通常容量，并遵循这里所述的标准流程工序。除非订单或请求进入承诺引擎130时指定有特殊加急服务，否则，该订单

或请求被认为（默认）是一个标准类型订单。如果订单或请求是标准类型的，那么它决不会被承诺为是特殊加急订单，除非它作为一个特殊加急订单被发送到承诺引擎130。换言之，承诺引擎130不会将一个标准请求流程转变成特殊请求流程，如针对加急订单所详述。

- 5 为了将该请求标识为普通请求，将针对标准类型订单在所有仓库中都建立和启用一个单独的通常容量物品。承诺引擎130在第一次查询APS应用程序之前将单独通常容量物品附加到标准类型订单中的每种物品中。这一通常容量物品具有每个仓库的独特供应图，并可用于所有的标准类型订单。承诺引擎130将知道应当附加通常容量物品，这是因为，除非在订单
- 10 中明确指示它是一个加急（佣金服务）订单，否则它将被默认为是一个标准类型订单。这种类型的订单无需包含其他参数来指示它是标准类型订单。

- 另一方面，加急订单是一种特殊订单和一种专门类型的有服务限制的订单或请求。加急订单具有独特的仓库截止时间和独特的容量提供（将在
- 15 后面作进一步的详述）。加急特殊订单或请求由客户和/或顾客运送条款来规定，并且是作为进入承诺引擎130的订单或请求，请求利用货运公司的通宵服务和仓库的佣金处理服务从仓库中发运出物品。其他所有订单和请求都被默认为标准类型订单（即，如果请求被输入时只有通宵服务，那么这是标准类型订单）。承诺引擎控制模块220的规则假定顾客在试图
- 20 使其成为加急请求之前先选择将请求作为标准通宵服务请求来提交，因为在理论上这将可以以低得多的费用提供相同的服务等级。如果该请求不能作为一个标准订单类型来满足，而顾客愿意支付佣金，那么，在接收到来自承诺引擎130的该请求不能被满足的响应后，顾客可以将该请求作为特殊加急订单重新提交。根据本发明的一个方面，指定加急订单或请求不保
- 25 证第二天交货。如果没有足够的容量或库存，那么，即使请求是加急订单，它也无法被满足。

- 为了将该请求标识为加急，将针对加急特殊类型订单在所有仓库中都建立和启用一个单独的加急容量物品。承诺引擎130（具体地说，承诺引擎控制模块220）在第一次查询APS应用程序之前将加急容量物品（或标识
- 30 符）附加到加急订单中的每种物品中，以判断哪个仓库可用于承诺或满足

该请求。承诺引擎130将知道要附加加急容量物品，这是因为，该订单从客户发来时指定有通宵服务和佣金处理服务，正是这些内容将订单定义为特殊加急订单。如果只指定了通宵服务，那么它将被默认为是标准类型服务，并且将使用通常容量物品。

5 B. 承诺和承诺范围

应当理解，承诺引擎130只承诺仓库中实际上存在的准备要发运的库存（四壁（four wall）承诺）。因此，返回给客户的承诺将基于库存和容量的可用性，其中，库存的可用性包括仓库中现有的库存。然而，承诺引擎130还可以针对客户120所建立的标准订货-交货时间（standard lead
10 time）来承诺库存，如果客户120需要这种订货-交货承诺的话。在客户建立了订货-交货时间的情况下，供应量被假定为是不定的，这是因为，承诺引擎130不清楚供应渠道。承诺引擎130还可以进一步伸延到供应链，并考虑针对还在向仓库或其他中转站运送的物品进行承诺—可以信任这种
15 可用性标记作为预先运送通知。因此，根据本发明的一个方面，承诺引擎130可以将预先运送通知（ASN）和购买订单（PO）视为合格库存。订单承诺的PO是指要被传送给仓库的可供应的购买订单。这些PO被客户所拥有并
20 进行处理。再者，承诺引擎130还可以使用重新承诺特性，其中，出货被保存并被检验，这样，变动的可用性条件（例如，已被承诺的物品过后被损坏或丢失）将导致请求的重新承诺。

20 承诺引擎130不对订单的实现进行排序或最优化（加急订单除外），如上面所提及并如下面所详述。因此，标准类型请求通常按先进先出（FIFO）方式进行处理。于是，承诺引擎130试图根据接收到订单的次序来保留订单，而与顾客的身份无关，并且与可能由同一顾客发出的其他订单也无关。例如，在第一顾客刚好在第二顾客发出订购一百万件小器具的
25 订单之前发出了订购10件小器具的订单的情况下，即使此时库存正好有1百万件小器具，承诺引擎130也将首先满足10件小器具的订单。因此，第二顾客的订单将不会因为其订购量大（或来自优先顾客）而比第一顾客的订单优先。在满足了第一订单后，承诺引擎130将不能承诺第二订单。然而，如果客户允许有多次出货，那么，承诺引擎130也可以将余下的小器
30 具发运给第二顾客，如果第二顾客有这种要求的话。当然，如上所述，承

诺引擎130可以承诺订购所有小器具的第二订单，如果客户已提交了使得订单交货可行的小器具订货-交货时间的话。

C. 用于判断仓库能否满足请求的承诺引擎规则

承诺引擎130（具体地说，承诺引擎模块240）包括若干条规则，承诺引擎130将利用这些规则，根据从APS应用程序160（具体地说，APS应用程序160的GATP功能）转发到承诺引擎130的数据，来判断仓库能否满足客户请求。如上所述，GATP功能保留了各仓库中的物品的画面，以及这些物品是否可以在具体日期之前从各仓库发出。如这里所提到，GATP搜索是承诺引擎130对APS应用程序160的查询，因此，承诺引擎130请求库存和仓库容量，以判断仓库能否满足客户请求。

承诺引擎控制模块240在查询APS应用程序160来实现GATP搜索以试图满足请求时作出了一些假定。第一，承诺引擎130假定保持单方出货比从首要仓库中寻找货源具有更高的优先级。换言之，承诺引擎130将试图由首要仓库来满足整个订单。如果这一仓库不能满足整个订单，那么将试图在所有次要仓库中满足整个订单。即使在首要仓库中发现有一部分可供货，也必须在搜索完所有其他次要仓库后，才能分割该订单。第二，对于加急订单，实现通宵交货比保持单方出货具有更高的优先级。换言之，更希望将出货量分割到一些仓库，以便通宵交货订单，而不是保持单方出货后再发运。

第三，对于有日期限制的订单，保持按时交货比保持单方出货具有更高的优先级。换言之，要求将出货量分割到一些仓库，以便在所请求的交货日期之前交付订单，而不是保持单方出货后再发运。第四，对于有日期限制的订单，保持单方出货的订单的最便宜的交货服务等级比从首要仓库中寻找货源具有更高的优先级。不过，这不适用于有服务和加急限制的订单（其中服务等级已有规定）。换言之，承诺引擎130将试图采用最便宜的交货服务从首要仓库来完全满足这一请求。如果首要仓库不能承诺这一请求，那么，承诺引擎130将指望使用最便宜交货服务的所有次要仓库。如果使用最便宜交货服务的任何次要仓库都不能完全满足这一请求，那么，首要仓库的优先级变高。不论需要什么样的交货服务，在转向次要仓库之前，承诺引擎130都将先由首要仓库来完全满足订单。此时，在搜索

完最便宜的服务后，首要仓库中的所有服务都优先于任何次要仓库中的所有服务。

第五，对所有请求类型的ATP搜索将查找与请求有关的库存物品和容量物品。如果在出货请求中牵扯到任何物品（无论是库存物品还是容量物品），那么，在同一日期和在同一仓库，ATP搜索必须共同满足这些物品。然而，请求仍然可以按日期或仓库被分割。如果请求（无论是单种物品或者多种物品）需要被分割，那么，当它被分割时，需要保持出货组的比率。此外，承诺引擎130必须能识别一些仓库的以及同一仓库的不同日期的若干个出货。可能需要最大数量的部分量(partial)，以限定出自单方仓库的出货数。

第六，承诺引擎130在这样的规则下操作：库存可以累积（如果它不被使用的话，它可以一天一天地滚动）而容量则不可累积。在这种假设下，库存可以以多种数量、在多个日期上找到，并且可用于在找到该库存的日期，甚至该日期之后，来满足单个或多个出货。发生这种情况时，必须保证全部容量需求可在该最后日期被满足，该日期便成为出货日。如何判定“出货日”的一个例子是：如果在日期（n）可从首要仓库得到库存，而在日期（n+1）首要仓库中的容量可用，那么，出货日应是日期（n+1）。

D. 实现承诺评估的可用性

当一个出货组（即要被发运的物品组）被发送到APS应用程序160的GATP应用程序（或服务器）时，ATP检查首先将试图在所请求的出货日期在首要仓库满足所有物品，这在调用GATP之前确定。如果在所请求的出货日期在首要仓库能满足整个出货组，那么数量和日期作为响应被发回。如果在所请求的日期和在首要仓库能满足请求，那么没有理由继续进行搜索，因为这是最佳响应。如果在供应范围内的某日在首要仓库中无法找到总的数量，那么，GATP将查看搜索规则所规定的每个次要仓库。如果在查看了每个仓库后仍无法找到总的数量，那么，总的数量可在标准订贷-交货时间日期由首要仓库当作不定供应来实现。

例如，如果在所请求的日期在首要仓库能满足订单的50%，而在所请求的出货日期后的5天内连续每天都能满足订单的10%，那么，GATP将在第五天（最后那天）返回100%。作为另一个例子，如果在所请求的日期

满足请求的0%，而在标准订货-交货时间内的不同日期能满足请求的100%，那么，该响应将包括单个日期和数量的组合，其中，含有找到供应能实现请求的最后日期，或者含有所订购的需求量已累积够用的日期。所找到的能满足请求的日期和部分量的组合可以从ATP的供应/需求图中获得。

- 5 如果不论为其找到的日期和部分量的组合如何，在首要仓库中都不能满足整个出货组，那么，该响应将在不定订货-交货供应存储体(bucket)中包括单个日期和数量（如果在次要仓库中不能满足总的数量）。即使它能按时或在标准订货-交货时间内满足订单的一部分，它也将推进到不定供应存储体。不过，如果客户允许分割出货量，那么，承诺引擎130将尝
- 10 试将出货组按仓库和日期来分割，以满足实际需求。来自订货-交货时间存储体内的日期和部分量组合（如能找到）可以从ATP的供应/需求图中获得。

- 在响应中没有返回的但能从ATP的供应/需求图中获得的数据可以被承诺引擎130用来确定部分出货量，假定客户允许这样的部分量。承诺引擎130的特定逻辑和选择过程将在各个单独的请求流程中进行描述，因为它们对请求类型来说是独特的。承诺引擎130根据所允许的最大出货数（ $2n+1$ 个）来选择部分量，这里 n 是用于这一请求的寻源规则中所规定的仓库数。当承诺引擎130利用供应/需求图来确定部分量的可用性时，承诺引擎130首先试图完全由寻源规则所规定的任何仓库来满足出货组。如果
- 20 承诺引擎130无法在订货-出货时间内满足整个出货组而客户允许分割，那么，承诺引擎130按以下方式分割出货量：根据需要，从各仓库中要求所请求的日期/数量；在最高等级的仓库中开始启动；然后，根据需要，从每个仓库中要求在供应范围内当累积到最大量时的日期上的累积可供
- 25 供应量；最后，要求任何剩余量，作为首要（最高等级）仓库的不定出货量。

- 如果在所请求的日期和在首要仓库认定了整个请求，那么ATP搜索将终止，因为这是最佳响应。如果在所请求的日期和在首要仓库没有认定整个请求，那么ATP搜索将通过首要仓库继续进行，并还将通过每个次要仓库继续进行。次要仓库将利用寻源规则在GATP内被预先确定和建立，这些规则告诉ATP搜索次要仓库时的次序。这种搜索与以上针对首要仓库所述
- 30 的搜索类似。如果在首要仓库（或在任何仓库）满足了总的数量，那么搜

索被终止。承诺引擎130不用计算所有可能的响应，因为这具有重要的性能牵连。承诺引擎130的功能性将在以下单独请求流程中作进一步的详细讨论。

E. 各请求类型的承诺流程

5 本节描述四种请求类型（无限制，有日期限制，有服务限制，和加急（这是有服务限制的特例））中的每种请求类型。如上所述，承诺引擎知道它所处理的请求的类型，根据它所属的请求类型，承诺引擎130将确定要被传送到GATP应用程序的所请求出货日期。

1. 无限制的请求

10 如果请求类型是无限制的，那么，发出请求时没有具体的日期或服务等级。承诺引擎将所请求的出货日期设置为当前日期，同时遵守下午3点（或预定的其他某个时间）为接受订单的截止时间。最好针对系统填入时间，以便在最终出货之前得到、承诺、形成和处理该订单。每个仓库都有
15 独立的截止时间（时区应适当）。因此，在每个仓库中都可以有不同的截止时间。

2. 有服务限制的请求

 如果请求类型是有服务级限制的，那么，发出请求时就有所要求的交货服务（即，通宵交货，地面服务）。如果是这种情况，那么承诺引擎将
20 请求出货日期设置为当前日期，同时遵守下午3点（或预定的其他某个时间）为接受订单的截止时间。需要针对系统填入时间，以便在最终出货之前得到、承诺、形成和处理该订单。每个仓库都有独立的截止时间（时区
 应适当）。因此，在每个仓库中都可以有不同的截止时间。

 有服务限制的请求的前提是：当服务级适合于出货并且系统假定服务级正确时，客户才发送服务级。例如，如果物品是不能通过空中服务发运
25 的危险品，那么系统依赖于客户来提出有效的发运方法。对于有服务限制
 的请求，将不会出现重量和体积检验。

3. 有日期限制的请求

 如果请求是有日期限制的，那么，当请求进入到承诺引擎时，承诺引擎
30 将知道客户是否使用了与本发明的承诺引擎和系统有关的货运公司，比如
 United Parcel Service, Inc (UPS)。如果客户使用了与该系统有关

货运公司，如UPS，那么货运公司可以接收交货日期请求。例如，如果可用的最便宜的服务是需要5天才能交货的地面服务，而所请求的交货日期在接收到请求后只有3天时间，那么，不能选择采用最便宜的服务。因此，当承诺引擎130调用交货网络以选择一种服务来启动搜索时，应当考虑到这一点。所用的请求出货日期是当前日期。另一方面，如果使用与系统无关的货运公司，那么，不将交货日期要求发给该货运公司，而承诺引擎130必须向客户请求出货日期。

4. 加急的请求

如果请求类型是加急订单，那么，客户/顾客将支付该订单的佣金，以便订单第二天就实现。客户将向系统提交：请求第二天（通宵）服务和佣金处理服务。佣金处理服务是为了将加急订单与其他所有标准类型的订单区别开。所有加急订单都需要加急容量。应当理解，加急容量将代表超过正常标准仓库截止时间直到该仓库的加急截止时间的时段。换言之，如果仓库的正常工作时间是从上午7点到下午3点，那么，仓库的标准通常容量将是每天8小时（或每天处理的x件标准类型订单）。如果加急截止时间为下午7点，那么，加急容量将代表下午3点到下午7点之间的小时数或容量（每天4小时或每天X件加急订单）。应当理解，加急请求总是涉及加急容量。容量概况将被密切监视，这样，必要时，容量可以以人工方式在通常容量与加急容量之间转换。

根据本发明的一个方面，针对任意一天能接受多少加急订单（加急容量），没有一个上限。如果超出这一最大限度，请求将被拒绝。由于系统可超过仓库的标准截止时间接受加急订单，所以加急订单可以有独自の截止时间。加急订单只对照现有库存而不是对照ASN或PO被承诺。如果现有库存不够用，那么将不返回承诺或返回一个无效承诺。由于这不是GATP中的标准功能性，因此，需要在承诺引擎中出现这种差异。

F. 承诺引擎的订单管理功能

订单管理过程包括订单形成、订单变动、订单取消和退回订单处理。这些销售订单（无论是交互式的还是非交互式的）从客户处被接收到并被传送到承诺引擎。承诺引擎130对照承诺对销售订单进行确认并将其发送到订单管理系统（OMS）以便进一步处理，然后，OMS又将订单发送到仓库

管理系统190以便实现。OMS可以是现成的常规组件（在图1的系统100中未示出），或者可作为另一系统组成部分的部件包含在系统100中。利用OMS，处理从客户进入系统的销售订单的大多数过程可以自动进行。

5 对于熟练技术人员而言，利用以上所述和附图中所给出的说明，可以设想出关于本发明的许多修改方式和其他实施方式。因此，应当理解，本发明并不局限于所公开的这些具体的实施方式，而还可以在附属权利要求书的范围内涉及这些修改方式和其他实施方式。尽管这里使用了特定术语，然而，它们只是从通用性和描述性意义上加以使用，而不是为了进行限定。

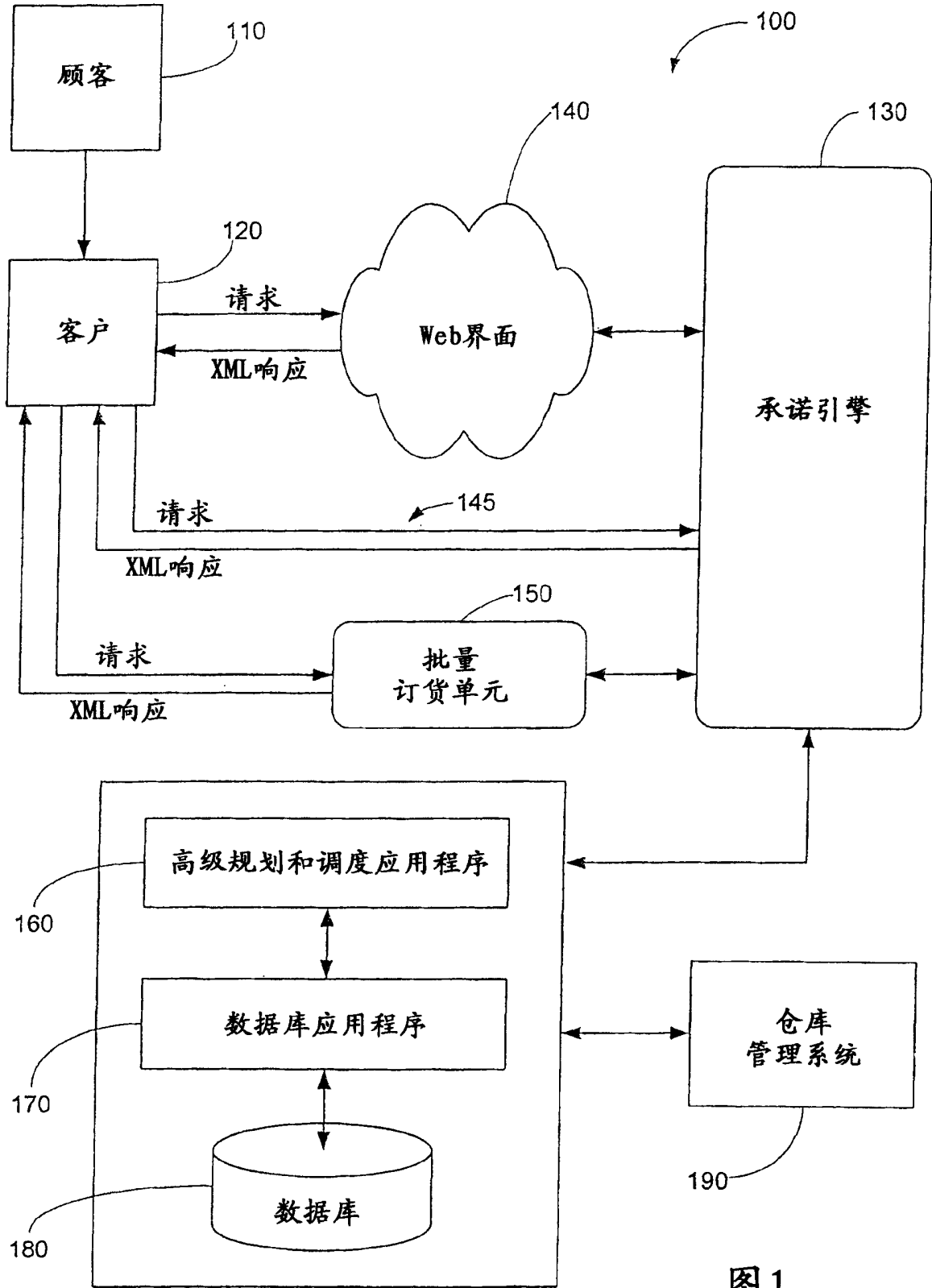


图1

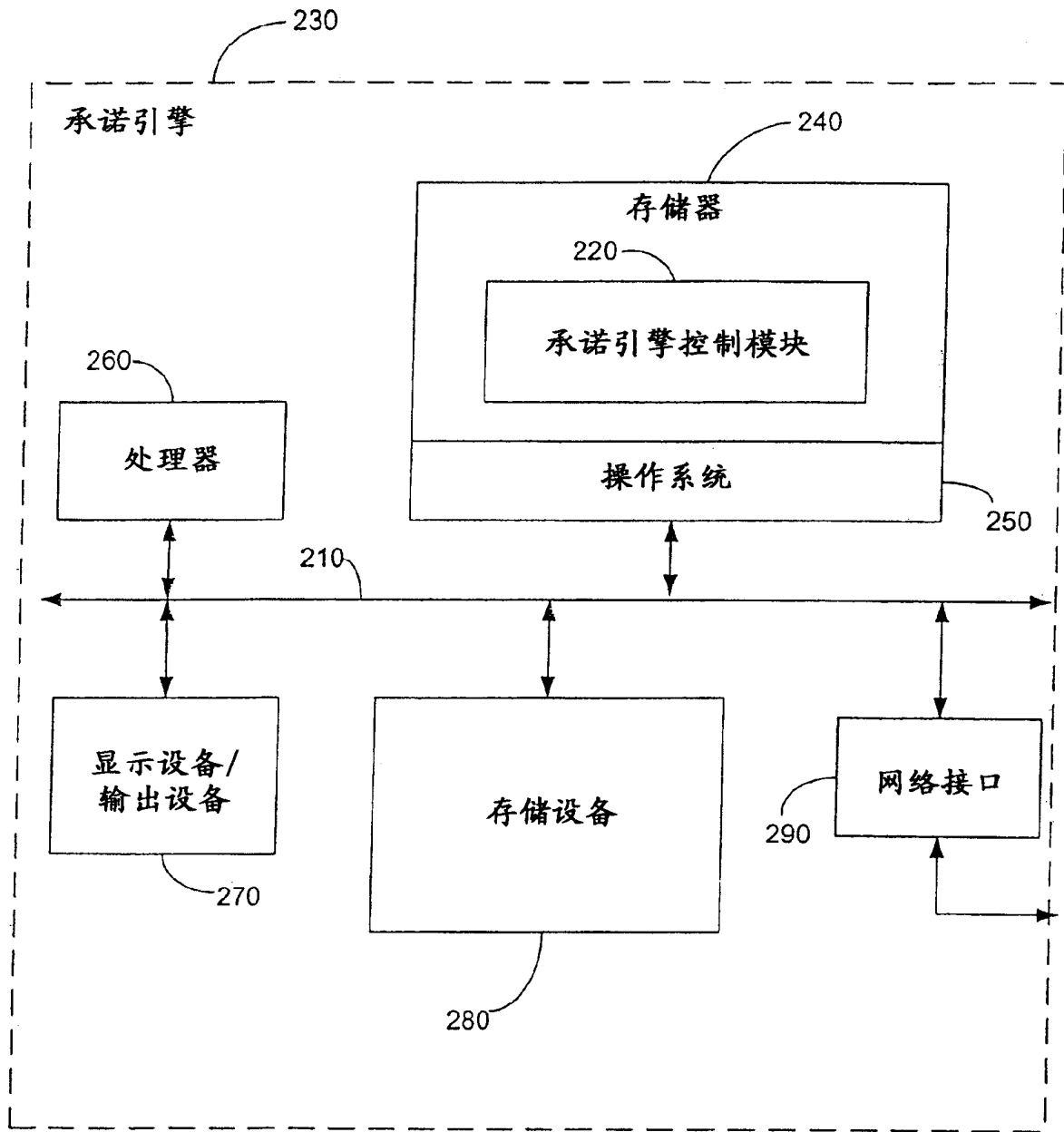


图2

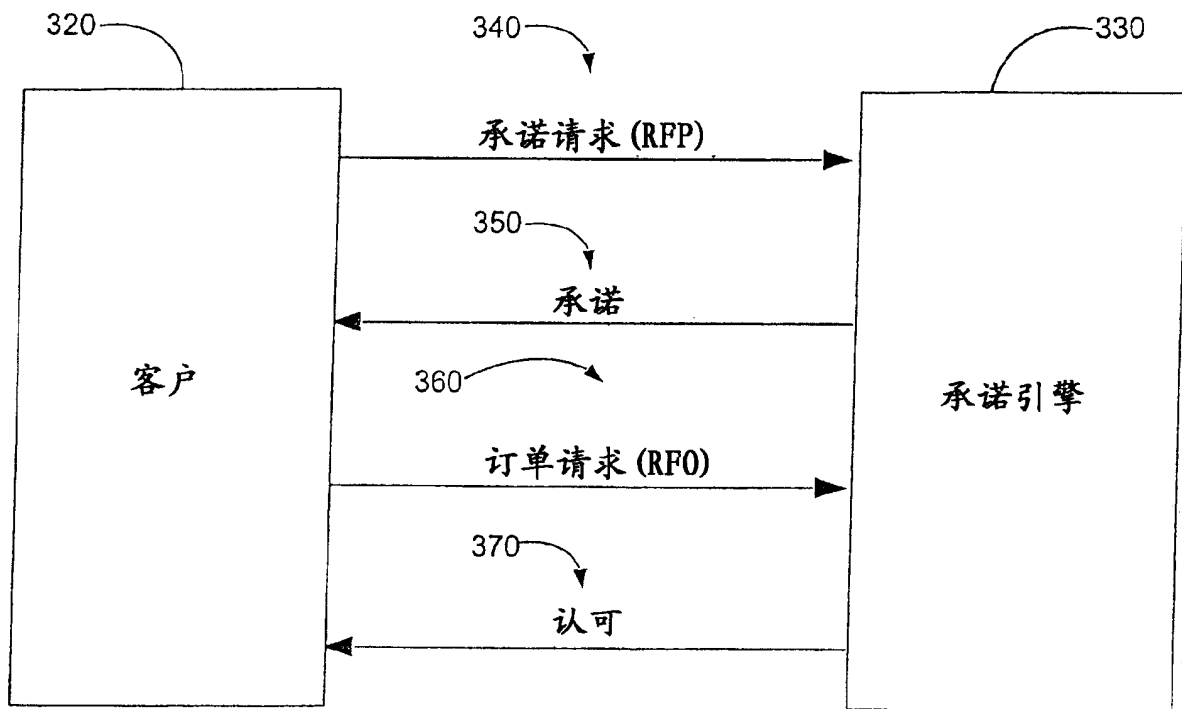


图 3

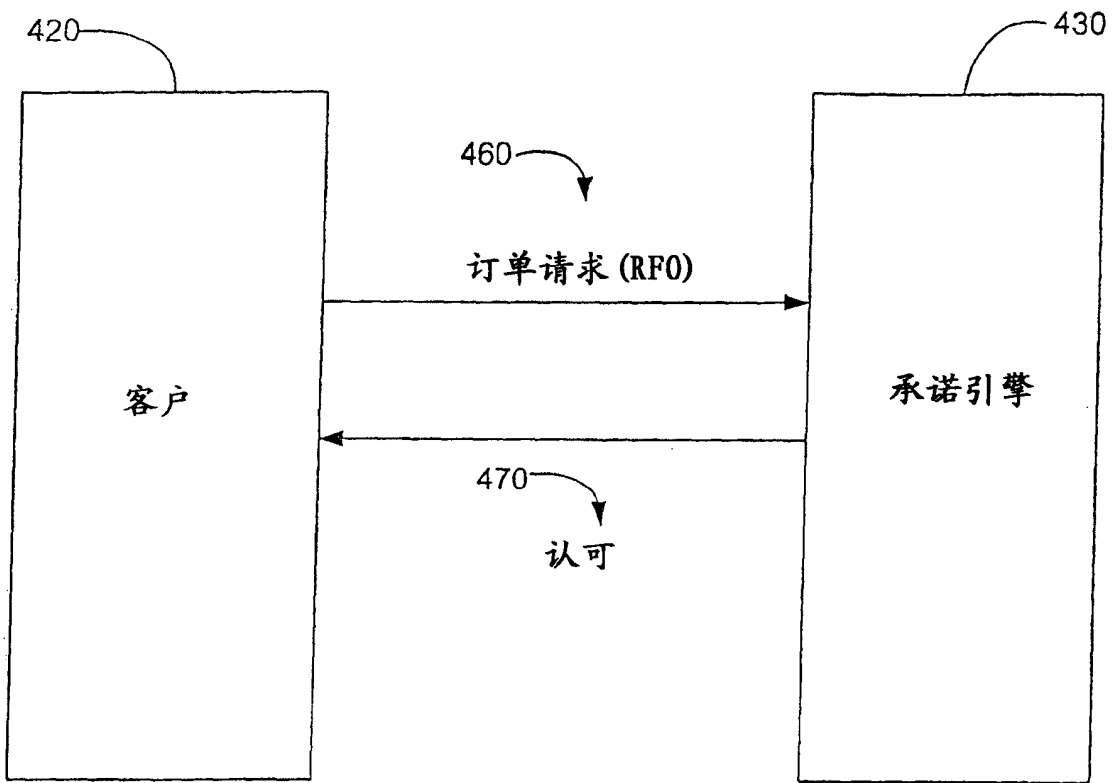


图4

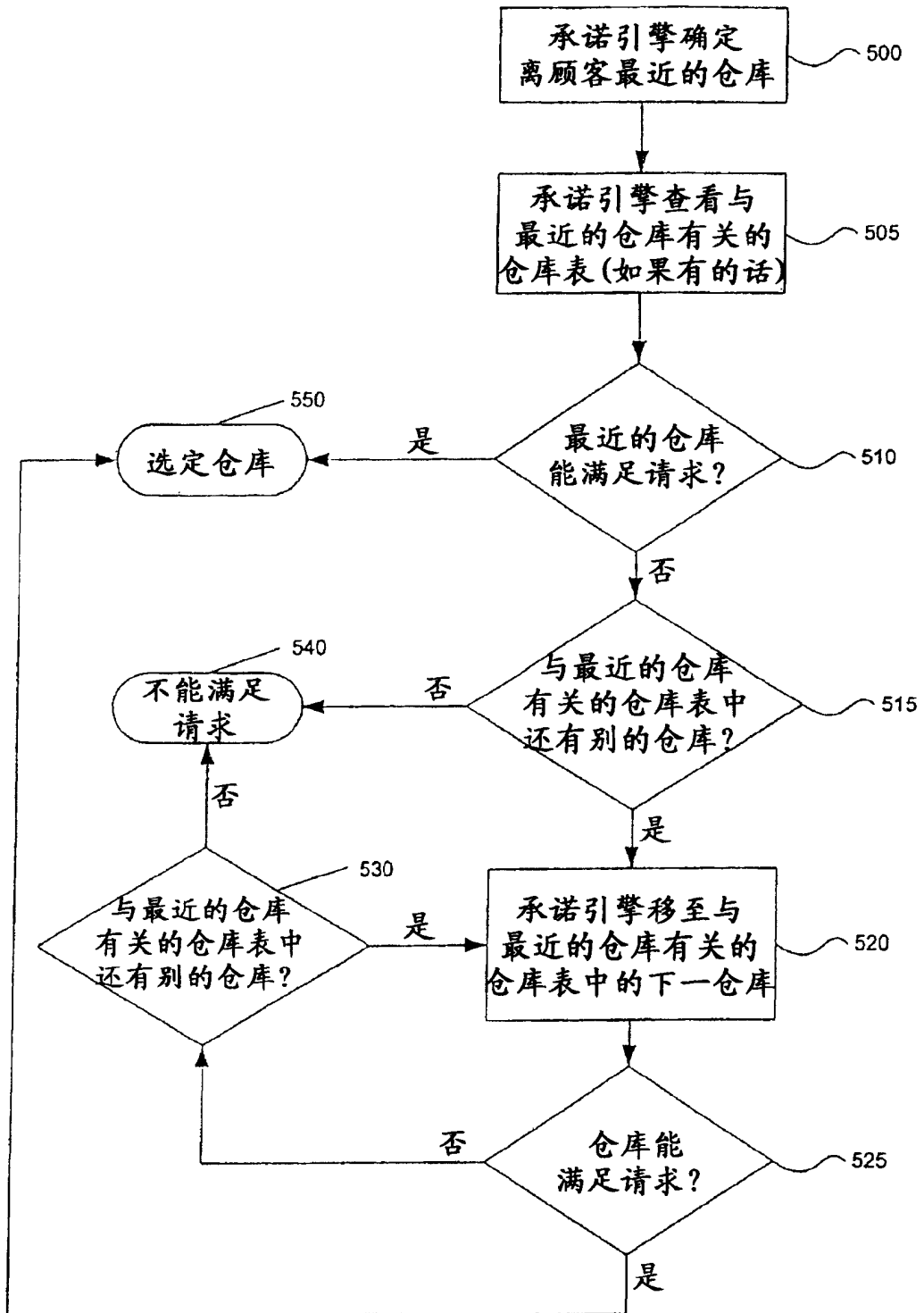


图 5A

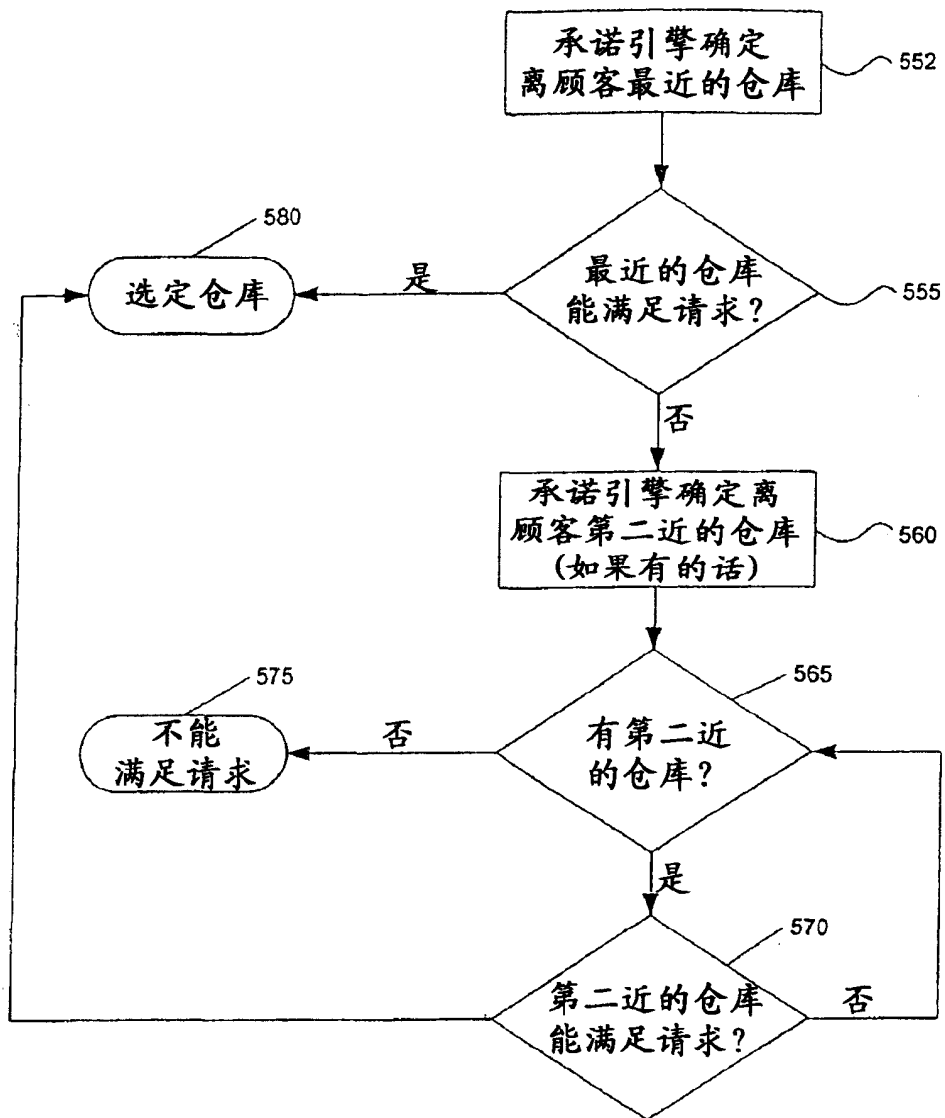


图 5B

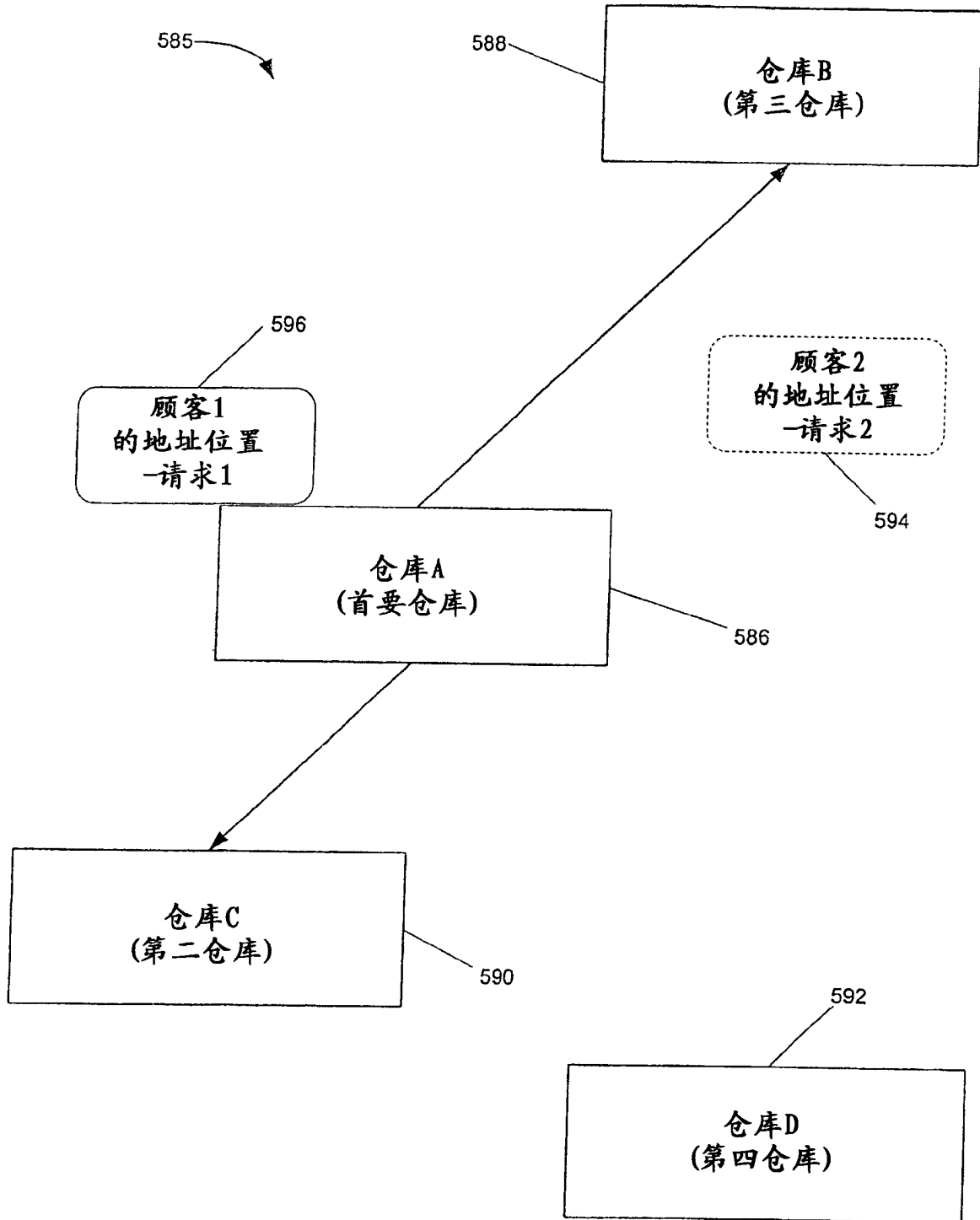


图 5C