



(12) 发明专利

(10) 授权公告号 CN 1647080 B

(45) 授权公告日 2010.04.14

(21) 申请号 03808943.2

(22) 申请日 2003.04.17

(30) 优先权数据

10/131,984 2002.04.25 US

(85) PCT申请进入国家阶段日

2004.10.21

(86) PCT申请的申请数据

PCT/GB2003/001674 2003.04.17

(87) PCT申请的公布数据

W02003/091829 EN 2003.11.06

(73) 专利权人 国际商业机器公司

地址 美国纽约

(72) 发明人 理查德·迪安·戴廷格

理查德·约瑟夫·史蒂文斯

(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 11038

代理人 李德山

(51) Int. Cl.

G06F 17/30 (2006.01)

(56) 对比文件

US 2001037345 A1, 2001.11.01, 说明书摘要, 第 49 - 51 段, 第 55 - 64 段.

US 2002032676 A1, 2002.03.14, 说明书第 65 段.

US 5121494 A, 1992.06.09, 全文.

审查员 田竞

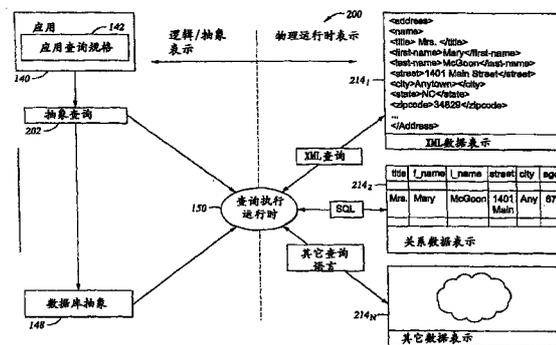
权利要求书 1 页 说明书 14 页 附图 8 页

(54) 发明名称

多数据库环境中存取数据的方法和计算机

(57) 摘要

本发明通常涉及以独立于数据被物理表示的特定方式的方式存取数据的系统、方法以及产品。在一个实施例中,数据库抽象层提供了独立于数据表示的特定方式的底层数据库的逻辑视图。在一个实施例中,数据库抽象层指定了数据库中数据的位置,以及用于存取数据的方法。还提供了基于数据仓库抽象层的查询抽象层。运行时部件将抽象查询转变为对于特定物理数据表示可以使用的形式。



1. 一种从多个分布式数据库中存取数据的方法,数据库中的至少一个使用与任何其他分布式数据库不同的物理数据表示,该方法包括:

从请求方实体接收根据请求方实体的查询规格的抽象查询;其中,查询规格定义了抽象查询的多个逻辑域,使得以独立于要存取的任何数据库的底层物理数据表示的方式定义抽象查询;

将抽象查询划分成多个子查询,子查询根据存取方法类型进行分组;以及

根据与特定子查询相关的存取方法,将每个子查询转换成与数据的特定物理数据表示一致的查询,其中每个存取方法能够通过定义每个物理实体的存取方法以及每个物理实体的位置规格,将逻辑域映射到数据的物理实体,所述位置规格标识数据库和所标识数据库中数据的位置。

2. 如权利要求 1 所述的方法,其中,从包括 SQL 查询类型,XML 查询类型和过程请求类型的组中选出存取方法类型。

3. 如任一在前权利要求所述的方法,其中,抽象查询包括至少一个选择标准和结果规格。

4. 如权利要求 1 或 2 所述的方法,还包括:

针对多个逻辑域的特定逻辑域的数据的物理实体,确定数据的物理实体是否位于本地高速缓存内;以及

如果数据的物理实体不在本地高速缓存内,存取由数据的物理实体的存取方法中的位置所指定的数据库。

5. 一种用于从多个分布式数据库存取数据的计算机,数据库中的至少一个使用与任何其他分布式数据库不同的物理数据表示,该计算机包括:

查询规格;

用于从请求方实体接收根据请求方实体的查询规格的抽象查询的装置,其中,查询规格定义抽象查询的多个逻辑域,使得以独立于要存取的任何数据库的底层物理数据表示的方式定义抽象查询;

用于将抽象查询划分成多个子查询的装置,其中子查询根据存取方法类型进行分组;

包括存取方法的数据库抽象部件,该存取方法通过定义每个物理实体的存取方法将逻辑域映射到数据的物理实体,数据库抽象部件还包括每个物理实体的位置规格,该位置规格标识数据库和数据在所标识数据库中的位置;以及

用于根据与特定子查询相关的存取方法,将每个子查询转换成与数据的特定物理数据表示一致的查询的运行时部件。

多数据库环境中存取数据的方法和计算机

技术领域

[0001] 本发明涉及用于在多数据库环境中存取数据的方法、计算机程序和计算机。

背景技术

[0002] 数据库是计算机化的信息存储和检索系统。关系数据库管理系统是使用关系技术存储和检索数据的计算机数据库管理系统 (DBMS)。最流行的数据库类型是关系型数据库, 即表格数据库, 其中数据被定义为能以各种不同的方式重组和存取。

[0003] 不考虑特定的数据结构, 在 DBMS 中, 请求方实体 (例如, 应用, 操作系统或者用户) 通过发出数据库存取请求要求存取指定的数据库。这种请求可能包括, 例如, 简单目录查找请求或者事务和事务组合, 其在数据库中读取、改变和增加指定的记录。通过使用高级查询语言生成这些请求, 例如结构化查询语言 (SQL)。说明性地, 使用 SQL 生成交互式查询, 用于从数据库中获得信息以及更新数据库, 例如国际商用机器公司 (IBM) 的 DB2, 微软的 SQL Server, 以及来自 Oracle, Sybase 和 Computer Associates 的数据库产品。术语“查询”是指一组用于从所存储的数据库中检索数据的命令。查询采用命令语言的形式, 其允许编程者和程序选择, 插入, 更新, 找出数据的位置, 等等。

[0004] 通常, 数据挖掘和数据库查询应用面临的一个难题是其与所给数据库模式 (例如, 关系型数据库模式) 的紧密关联。此关联使得当相应的底层数据库模式发生改变时很难支持应用。另外, 限制了应用向可选底层数据表示的迁移。在当前环境下, 上述缺点大多是由于应用对 SQL 的依赖引起的, SQL 假定所查询的信息是使用关系模型表示的。此外, 因为在 SQL 查询表示内引用了特定的数据库表, 列和关系, 因此, 所给定的 SQL 查询依赖于特定的关系模式。结果, 这些限制引发了许多难题。

[0005] 一个难题在于, 在底层关系数据模型中的改变需要对 SQL 基础进行改变, 而应用是基于 SQL 基础构建的。因此, 应用设计者必须或者放弃改变底层数据模型以避免应用维护, 或者必须改变应用以反映底层关系模型的改变。另一个难题在于, 扩展应用以便与多个关系数据模型协同工作需要分别的应用版本, 以反映通过每个独特关系模式驱动的独特 SQL 需求。然而, 因为 SQL 是被设计为用于关系系统, 另一个难题是如何使应用演变以便与可选数据表示协同工作。扩展应用以支持可选数据表示, 例如 XML, 需要重写应用的数据管理层, 以便使用非 SQL 数据存取方法。

[0006] 针对上述问题通常所使用的解决方法是软件封装。软件封装涉及使用软件接口或者部件来封装针对特定底层数据表示的存取方法。企业 JavaBean™ (EJB) 规范是一个示例, 企业 JavaBean™ (EJB) 规范是 Java™ 2 企业版 (J2EE) 套件技术的部件。在 EJB 实例中, 实体豆 (entity bean) 封装所给的数据集, 公开一组应用程序接口 (API), 所述 API 可以用于存取此信息。这是高度专业化的解决方法, 每当要存取一组新的数据, 或者当希望获得新的数据存取模式时, 就需要编写软件 (以新实体 EJB 的形式)。EJB 模型还需要代码更新, 应用构建和部署周期, 以便对底层物理数据模型的重组作出反应, 或者支持可选的数据表示。因为涉及更高级的 Java 编程技术, EJB 编程还需要专业化的技能。因此, EJB 解决方案以及

其他的相似解决方案很不灵活,并且对于维护用于存取演进的物理数据模型的通用查询应用而言费用昂贵。(Java 和所有基于 Java 的商标是 Sun 微系统公司在美国、其他国家或二者的商标)。

[0007] 除了存取不同类型数据表示的难题外,现今的环境由于数据常常是高度分布式的事实而变得复杂。类似因特网的流行基础架构包含数据源主机,其必须对于用户可访问以便体现价值。处理本地化、同质化数据的传统解决方案不再有效,而开发处理分布式和不同类型数据的解决方案尚有问题,因为这样的解决方案必须知道每个数据源的位置并且必须提供独特的逻辑(软件)以处理每个不同类型的数据表示。结果,通常的解决方案(例如提供数据仓库,其包含使用数据仓库的应用所需要的所有信息)很难适应正被使用的数据的位置或表示的改变,并且很难被重新部署以便与不同数据拓扑协同工作。当需要用附加的,公开可获得的信息扩充仓库内容时,数据仓库还引发了问题。在一些实例中,外部的数据源可能很大并且易于变化。在所给数据仓库中维护这种数据的本地拷贝,费用会是非常昂贵的。

[0008] 因此,需要一种改进的并且更加灵活的存取数据的方法,其不限于表示底层物理数据的特定方式。

发明内容

[0009] 根据一个方面,本发明提供了在多数据库环境中存取数据的方法,包括:从请求方实体接收根据请求方实体查询规格的抽象查询;其中查询规格提供抽象查询的多个逻辑域的定义;以及根据存取方法将抽象查询转换成与数据的特定物理数据表示一致的查询,该存取方法通过定义存取每个物理实体的方法以及每个物理实体的位置,将逻辑域映射到数据的物理实体。

[0010] 当然,可以使用计算机软件执行该方法。

[0011] 最好是通过数据模式和查询抽象提供用于分布式数据源的远程数据存取和集成的方法、计算机程序和计算机。

[0012] 本发明优选涉及以独立于物理表示数据的特定方式的方式存取数据的方法、计算机和计算机程序。优选地,提供抽象层以表示应用可使用的各种分布式数据源,并描述应用用以存取和/或者更新在这些数据源中所包含的信息的查询。优选地,运行时部件负责使用包含在数据库抽象部件(抽象层之一)中的信息,将抽象查询解析为针对一个或多个数据库的具体数据存取请求。

[0013] 一个实施例提供了存取具有特定物理数据表示的数据的方法。该方法包括对请求方实体提供查询规格,该查询规格包括用于定义抽象查询的多个逻辑域;以及提供数据库抽象,其将多个逻辑域映射到数据的物理实体。在一个实施例中,对于每个逻辑域,数据库抽象包括定义了数据物理实体位置的至少一个定位符,以及定义访问数据的物理实体的机制的存取方法。

[0014] 一个实施例提供了计算机,包括:处理器和存储器,该存储器至少包含(i)包括查询规格的请求方实体,该查询规格定义包括多个逻辑域的抽象查询;(ii)包括映射规则的数据库抽象部件,该映射规则将逻辑域映射到数据的物理实体,其中映射规则包括抽象查询的至少部分逻辑域的每个的位置规格,并且其中每个位置规格规定包含要存取的物理实

体的数据源的位置；以及 (iii) 运行时部件,用于根据映射规则将抽象查询转换成与数据的物理实体一致的查询。

[0015] 根据另一个方面,提供了计算机,包括:存储器,该存储器包含至少 (i) 提供包括多个逻辑域的抽象查询的定义的查询规格,(ii) 包括映射规则的数据库抽象部件,该映射规则将逻辑域映射到数据的物理实体,其中映射规则包括抽象查询的至少部分逻辑域的每个的位置规格,并且其中每个位置规格规定包含要存储的物理实体的数据源的位置;以及 (iii) 运行时部件,用于根据映射规则将抽象查询转换成与数据的物理实体一致的查询;以及适合执行存储器的内容的处理器。

[0016] 根据一个实施例,提供了在多个数据库环境中存取数据的方法,包括:对请求方实体提供包括用于定义抽象查询的多个逻辑域的查询规格;以及对于多个逻辑域中的每一个提供存取方法,该存取方法至少指定用于存取数据的方法以及数据的位置。

[0017] 优选地,该方法还包括,由请求方实体发出根据查询规格的抽象查询;将抽象查询转换为与数据的特定物理数据表示一致的查询;以及在针对多个逻辑域的特定逻辑域的数据物理实体的存取方法中访问该位置所指定的数据库。

[0018] 优选地,与特定物理数据表示一致的查询是 SQL 查询,XML 查询以及过程请求中的一个。

[0019] 优选地,将抽象查询转换成与特定数据物理表示一致的查询包括将抽象查询划分为根据存取方法类型分组的子查询。

[0020] 优选地,从包括 SQL 查询类型,XML 查询类型以及过程请求类型的组中选择存取方法类型。

[0021] 根据优选实施例,提供了在多个数据库环境中存取数据的方法,包括:由请求方实体发出根据请求方实体的查询规格的抽象查询;其中查询规格提供抽象查询的多个逻辑域的定义;以及根据存取方法将抽象查询转换成与数据的特定物理数据表示一致的查询,该存取方法通过定义每个物理实体的存取方法以及每个物理实体的位置,将逻辑域映射到数据物理实体。

[0022] 优选地,能够访问针对多个逻辑域的特定逻辑域的数据物理实体的位置所指定的数据库。

[0023] 优选地,抽象查询包括至少一个选择标准和结果规格。

[0024] 该方法最好包括针对多个逻辑域的特定逻辑域的数据物理实体,确定数据物理实体是否位于本地高速缓存中;如果不在本地高速缓存中,则通过数据物理实体的存取方法访问该位置所指定的数据库。

[0025] 优选地,抽象查询到与特定数据物理表示一致的查询的转换包括将抽象查询划分为根据存取方法类型分组的子查询。

[0026] 优选地,从包括 SQL 查询类型,XML 查询类型以及过程请求类型的组中选择存取方法类型。

[0027] 根据优选实施例,提供了包含程序的计算机可读介质,当处理器执行程序时,该程序执行在多个数据库环境中访问数据的操作,该程序包括:请求方实体的查询规格,该查询规格包括用于定义抽象查询的多个逻辑域;以及每个逻辑域的存取方法,每个逻辑域定义了存取数据物理实体的方法,以及被传递到用于存取物理实体的方法的多个参数,其中至

少一个参数是指定包含物理实体的数据源的位置的位置参数。

[0028] 例如,请求方实体可以是应用程序。

[0029] 优选地,多个存取方法中的每一个定义数据的特定物理表示和相应物理实体的位置。

[0030] 优选地,操作包括:请求方实体发出根据查询规格的抽象查询;将抽象查询转换成与特定物理数据表示一致的查询;并且访问由针对多个逻辑域的特定逻辑域的数据物理实体的位置所指定的数据库。

[0031] 根据优选实施例,提供了包含程序的计算机可读介质,当处理器执行程序时,该程序执行存取具有特定物理数据表示的数据的操作,该操作包括:由请求方实体发出根据请求方实体的查询规格的抽象查询;其中查询规格定义了抽象查询的逻辑域;并且根据存取方法将抽象查询转换成与数据的特定物理数据表示一致的查询,该存取方法通过为每个物理实体至少定义用于存取物理实体的方法和物理实体的位置,将逻辑域映射到数据物理实体。

[0032] 根据优选实施例,提供了计算机,包括:存储器,该存储器至少包含(i)包括查询规格的请求方实体,该查询规格定义抽象查询,该抽象查询包括多个逻辑域;(ii)包括映射规则的数据库抽象部件,映射规则将逻辑域映射到数据的物理实体,其中映射规则包括抽象查询的至少部分逻辑域的每个的位置规格,并且其中每个位置规格指定包含要存取的物理实体的数据源的位置;以及(iii)运行时部件,用于根据映射规则将抽象查询转换成与数据的物理实体一致的查询;以及适合于执行存储器的内容的处理器。

[0033] 优选地,由相应位置规格指定的数据源的第一部分是本地的,且第二部分是远程的。

附图说明

[0034] 现在将参考以下附图,通过示例的方式说明本发明的优选实施例;

[0035] 图1是根据本发明的优选实施例图解使用的计算机系统;

[0036] 图2A图解了软件部件的关系视图;

[0037] 图2B是用于关系数据存取的数据库抽象和抽象查询的一个实施例;

[0038] 图3根据本发明的一个实施例,图解了运行时部件操作的流程图;

[0039] 图4根据本发明的一个实施例,图解了运行时部件操作的流程图;

[0040] 图4图解了软件部件的关系视图,在其中可存取多个数据源;

[0041] 图6图解示出了包括多个逻辑域的抽象查询602;

[0042] 图7图解了以关系存取方法配置的数据库抽象部件的域规格;以及

[0043] 图8图解了以过程存取方法配置的数据库抽象部件的域规格。

具体实施方式

[0044] 介绍

[0045] 本发明通常涉及一种系统、方法以及产品,用于以独立于物理表示数据的特定方式的方式存取数据。数据可以包括多个不同数据源。

[0046] 在一个实施例中,数据库抽象层提供独立于数据表示的特定方式的一个或多个底

层数据库的逻辑视图。当有多个数据源时,用标识要存取的数据的位置的位置规格配置数据库抽象层的实例。还提供了基于数据库抽象层的查询抽象层。运行时部件将抽象查询(根据查询抽象层构造)转换为能够针对特定数据物理实体使用的形式。

[0047] 本发明的一个实施例被实现为用于例如下述图 1 中所示的计算机系统 100 的计算机系统的程序产品。程序产品的程序定义了实施例的功能(包括这里所述的方法),并且能包含在各种信号承载介质上。图解的信号承载介质包括但不限于:(i)在不可写存储介质上永久存储的信息(例如,计算机中的只读存储器设备,例如 CD-ROM 驱动器可读取的 CD-ROM 盘);(ii)在可写存储介质上存储的可改写信息(例如,磁盘驱动器中的软盘或者硬盘驱动器);或者(iii)通过通信介质传给计算机的信息,例如,通过计算机或者电话网络,包括无线通信。后者实施例包括从因特网和其他网络下载的信息。

[0048] 通常,被执行以实现本发明实施例的程序可以是部分操作系统或特定应用程序,部件,程序,模块,对象或者指令序列。优选实施例的软件通常包括大量指令,该指令被本地计算机转换成机器可读格式、因此可执行的指令。此外,程序由变量和数据结构组成,其或者本地驻留到程序中,或者可在存储器或存储设备中可找到。然而,应当知道,下面任何特殊的术语仅仅是为了方便,因此本发明不应限于仅仅使用于这样的术语所标识和/或所暗示的任何特定的应用中。

[0049] 环境的物理视图

[0050] 图 1 示出了可在其中实施本发明实施例的网络系统 100 的模块图。通常,网络系统 100 包括客户(例如,用户的)计算机 102(示出了三个这样的客户计算机)和至少一个服务器 104(示出了一个这样的服务器)。客户计算机 102 和服务器计算机 104 经由网络 126 相连。通常,网络 126 可以是本地局域网(LAN)和/或广域网(WAN)。在特定实施例中,网络 126 是因特网。

[0051] 客户计算机 102 包括经由总线 126 连接到存储器 112 的中央处理单元(CPU)110,存储设备 114,输入设备 116,输出设备 119,和网络接口设备 118。输入设备 116 可以是为客户计算机 102 提供输入的任何设备。例如,键盘,袖珍键盘,光笔,触摸屏,轨迹球,或者语音识别单元,音频/视频播放器等等可被使用。输出设备 119 可以是为用户提供输出的任何设备。例如,任何传统的显示屏。尽管所示与输入设备 116 分开,然而输出设备 119 可以与输入设备 116 结合。例如,可以使用具有集成触摸屏的显示屏,具有集成键盘的显示器,或者结合有文本语音转换器的语音识别单元。

[0052] 网络接口设备 118 可以是任何入口/出口设备,其被配置为容许客户计算机 102 和服务器计算机 104 之间经由网络 126 进行网络通信。例如,网络接口设备 118 可以是网络适配器或者是其他的网络接口卡(NIC)。

[0053] 存储设备 114 最好是直接存取存储设备(DASD)。尽管以单一单元示出,然而其可以是固定的和/或可移动的存储设备的结合,例如固定磁盘驱动器,软盘驱动器,磁带驱动器,可移动存储卡,或者光存储设备。存储器 112 和存储设备 114 可以是横越多个主要和辅助存储设备的一个虚拟位置空间的一部分。

[0054] 存储器 112 最好是足够大的随机存取存储器,以保存本发明优选实施例的编程和数据结构。虽然以单一实体示出存储器 112,然而应当理解,存储器 112 实际上可以包括多个模块,并且存储器 112 可以在多个级别上存在,从高速寄存器和高速缓存到较低速然而

较大的动态随机存取存储器芯片。

[0055] 图解性地,存储器 112 包含操作系统 124。说明性地,可以使用以产生良好效果的操作系统包括 Linux 和微软的 **Windows®**。更普遍地,可以使用支持在这里所公开功能的任何操作系统。

[0056] 还示出了包含浏览器程序 122 的存储器 112,当在 CPU 110 上执行时,浏览器程序支持在各种服务器 104 之间的导向和将网络地址定位于一个或多个服务器 104。在一个实施例中,浏览器程序 122 包括基于 Web 的图形用户接口 (GUI), GUI 容许用户显示超文本标记语言 (HTML) 信息。然而,更普遍地,浏览器程序 122 可以是能呈现从服务器 104 传送的信息的任何基于 GUI 的程序。

[0057] 可以以与客户计算机 102 相似的方式物理地安排服务器计算机 104。因此,所示服务器计算机 104 通常包括通过总线 136 彼此相连的 CPU 130,存储器 132 和存储设备 134。存储器 132 可以容量足够大以保存本发明优选实施例的编程和数据结构的随机存取存储器,其位于服务器计算机 104 上。

[0058] 服务器计算机 104 通常受存储器 132 中所示操作系统 138 的控制。示例操作系统 138 包括 **IBM® OS/400®**, **UNIX®**, **Microsoft® Windows®** 等等。更普遍地,可以使用能支持在这里所说明功能的任何操作系统 (IBM 和 OS/400 是国际商用机器公司在美国的商标;UNIX 是开放组织在美国和其他国家的注册商标;Microsoft 和 Windows 是微软公司在美国,其他国家或者双方的商标)。

[0059] 存储器 132 还包括一个或多个应用 140 和抽象查询接口 146。应用 140 和抽象查询接口 146 是包含多个指令的软件产品,所述指令在不同时期居于计算机系统 100 的不同存储器和存储设备中。当被服务器 104 中的一个或多个处理器读取并执行时,应用 140 和抽象查询接口 146 导致计算机系统 100 执行必需的步骤,以便执行体现本发明优选实施例各方面的步骤或者要素。应用 140 (更普遍地,任何请求方实体,包括操作系统 138 和,最高级而言,用户) 针对数据库发送查询。所发送的查询可以包括,例如,本地数据库 1561... 156N 和远程数据库 1571... 157N,全体被称为数据库 156-157。图解性地,所示数据库 156 是存储设备 134 中数据库管理系统 (DBMS) 154 的一部分。更普遍地,在这里所使用的术语“数据库”是指不考虑特定物理表示的任何数据的集合。通过图解的方式,可以根据关系模式 (可通过 SQL 查询存取) 或 XML 模式 (可通过 XML 查询存取) 组织数据库 156-157。然而,本发明不限于特定的模式,并预期扩展到当前未知的模式。如在这里所使用的,术语“模式”通常指数据的特定排列。

[0060] 在一个实施例中,根据每个应用 140 所包括的应用查询规格 142,定义应用 140 发出的查询。应用 140 发送的查询可以是预先定义的 (即,作为应用 140 一部分的硬件编码),或者是响应输入 (例如,用户输入) 生成的。在两者中的任何一种情况,使用抽象查询接口 146 定义的逻辑域构造查询 (在这里称之为“抽象查询”)。尤其是,通过抽象查询接口 146 的数据库抽象部件 148 定义在抽象查询中所使用的逻辑域。运行时部件 150 执行抽象查询,其将抽象查询转换成与一个或多个数据库 156-157 中所包含的数据的物理表示一致的形式。参照图 2A-B 进一步说明应用查询规格 142 和抽象查询接口 146。

[0061] 在一个实施例中,用户通过图形用户接口 (GUI) 规定查询要素。应用 140 生成 GUI 的内容。在特定实施例中,GUI 的内容是超文本标记语言 (HTML) 内容,其可以在具有浏览

器程序 122 的客户计算机系统 102 上呈现。因此,存储器 132 包括适合服务来自客户计算机 102 和请求的超文本传输协议 (http) 服务器进程 138(例如,web 服务器)。例如,进程 138 会响应存取数据库 156 的请求,数据库 156 图解性地地位于服务器 104 上。针对来自数据库 156-157 的数据的传入客户机请求启动应用 140。当由处理器 130 执行时,应用 140 使服务器计算机 104 执行体现本发明优选实施例各个方面的步骤或者要素,包括存取数据库 156-157。在一个实施例中,应用 140 包括多个被设置为构造 GUI 单元的 servlet,之后由浏览器程序 122 呈现 servlet。在经由应用 140 存取远程数据库 157 的情况下,用位置规格配置数据库抽象部件 148,该位置规格标识包含要检索的数据的数据库。以下将更具体描述此最后一实施例。

[0062] 图 1 只是网络相连的客户计算机 102 和服务器计算机 104 的一个硬件 / 软件配置。本发明的实施例能应用到任何类似的硬件配置,而不必考虑计算机系统是否复杂,是否多用户计算设备,单用户工作站,或者没有自身的非易失存储设备的网络设备。另外,可以理解,虽然提及特定标记语言(包括 HTML),然而本发明不限于特定语言,标准或者版本。因此,本领域的技术人员会认识到,本发明既适合于其他标记语言,也适合于非标记语言,并且,本发明还适合于特定标记语言将来的改变,以及当前未知的其他语言。同样地,图 1 中所示的 http 服务器进程 138 只是示意性的,本发明也考虑到适合支持任何已知和未知协议的其他实施例。

[0063] 环境的逻辑 / 实时视图

[0064] 根据本优选实施例,图 2A-B 示出了发明的多个相关部件。请求方实体(例如,一个应用 140)发出由请求方实体的相应应用查询规格 142 定义的请求 202。结果查询 202 在这里通常被称为“抽象查询”,因为查询是根据抽象(例如,逻辑)域,而不是通过直接引用数据库 156-157 中的底层物理数据实体来构造的。结果,可以定义独立于所使用的特定底层数据表示的查询。在一个实施例中,应用查询规格 142 包括用于数据选择的标准(选择标准 204),以及根据选择标准 204 要返回的域的显式规格(返回数据规格 206)。

[0065] 由数据库抽象部件 148 定义由应用查询规格 142 指定并用于构造抽象查询 202 的逻辑域。通常,数据库抽象部件 148 将信息公开为可用于由应用 140 发出的查询(例如,抽象查询 202)的一组逻辑域,以规定数据选择标准,以及规定查询操作回送的结果数据的形式。以独立于数据库 156-157 中正使用的底层数据表示的方式定义逻辑域,由此,容许要形成的被松耦合到底层数据表示的查询。

[0066] 通常,数据库抽象部件 148 包括多个域规格 $208_1, 208_2, 208_3, 208_4$ 和 208_5 , (作为示例,示出了五个),合称为域规格 208。特别地,为每个可用于构造抽象查询的逻辑域提供域规格。每个域规格包括逻辑域名 $210_1, 210_2, 210_3, 210_4$ 和 210_5 (合称为域名 210) 和相关的存取方法 $212_1, 212_2, 212_3, 212_4$ 和 212_5 (合称为存取方法 212)。存取方法将逻辑域名关联(即,映射)到数据库中(例如,数据库 156 之一)的特定物理数据表示 $214_1, 214_2, \dots, 214_n$ 。通过图解的方式,示出了两个数据表示,即 XML 数据表示 214_1 和关系数据表示 214_2 。然而,物理数据表示 214_n 指明考虑到了任何已知或未知的其他数据表示。

[0067] 根据要支持的逻辑域的不同类型的数量,考虑任意数量的存取方法。在一个实施例中,提供了存取简单域,过滤域和合成域的方法。域规格 $208_1, 208_2$ 和 208_5 分别示例了简单域存取方法 $212_1, 212_2$ 和 212_5 。简单域被直接映射到底层物理数据表示的特定实体(例

如,映射到所给数据库表和列的域)。通过图解的方式,在图 2B 中所示的简单域存取方法 212₁ 将逻辑域名 210₁ (“FirstName”) 映射到名为“contact”的表中名称为“f_name”的列。域规格 208₃ 例示了过滤域存取方法 212₃。过滤域标识相关物理实体并提供用于在物理数据表示内定义特定项子集的规则。在图 2B 所提供的示例中,过滤域存取方法 212₃ 将逻辑域名 210₃ (“AnyTownLastName”) 映射到名为“contact”的表中名称为“l_name”的列中的物理实体,并为城市 Anytown 中的个人定义了过滤器。过滤域的另一个示例是 New York ZIP 码域,其映射到 ZIP 码的物理表示,并将数据唯一限定到针对纽约州定义的那些 ZIP 码。域规格 208₄ 例示了合成域存取方法 212₄。合成存取方法使用作为存取方法定义的一部分提供的表达式由一个或多个物理域计算逻辑域。这样,可以计算出底层物理数据表示中不存在的信息。在图 2B 所图解的示例中,合成域存取方法 212₃ 将逻辑域名 210₃ “AgeIndecades” 映射到“AgeInYears/10”。另一个示例是销售税域,其通过将销售价格域乘以销售税率而合成。

[0068] 本发明考虑到底层数据的任意给定数据类型的格式(例如,日期,十进制数等等)也许各不相同的情况。因此,在一个实施例中,域规格 208 包括类型属性,其反映了底层数据的格式。然而,在另一个实施例中,域规格 208 的数据格式与相关底层物理数据不同,在此情况下,存取方法负责以请求方实体所假定的适当格式返回数据。于是,存取方法最好应知道假定的数据格式(即,根据逻辑域),以及底层物理数据的实际格式。然后,存取方法能将底层物理数据转换为逻辑域的格式。

[0069] 通过示例,图 2 中所示的数据库抽象部件 148 的域规格 208 代表被映射到关系数据表示 212₂ 中表示的数据的逻辑域。然而,数据库抽象部件 148 的其他实例将逻辑域映射到其他的物理数据,例如 XML。另外,在一个实施例中,以针对过程数据表示的存取方法配置数据库抽象部件 148。下面参照图 8 说明这种数据库抽象部件 148 的一个实施例。

[0070] 以下,在表 I 中图解示出了对应于图 2 中所示的抽象查询 202 的抽象查询。通过图解方式,使用 XML 定义数据库抽象 148。然而,可以使用其他任何有效的语言。

[0071] 表 I 查询示例

[0072] 001<? xml version = " 1.0 " ? >

[0073] 002<!--Query string representation : (FirstName = " Mary " AND LastName =

[0074] 003 " McGoon ")OR State = " NC " -->

[0075] 004<QueryAbstraction>

[0076] 005<Selection>

[0077] 006<Condition internal I D = " 4 " >

[0078] 007<Condition field = " FirstName " operator = " EQ " value = " Mary "

[0079] 008 internalID = " 1 " />

[0080] 009<Condition field = " LastName " operator = " EQ " value = " McGoon "

[0081] 010 internalID = " 3 " relOperator = " AND " ></Condition>

[0082] 011</Condition>

[0083] 012<Condition field = " State " operator = " EQ " value = " NC " internalID = " 2 "

[0084] 013 relOperator = " OR " ></Condition>

```
[0085] 014</Selection>
[0086] 015<Results>
[0087] 016<Field name = " FirstName " />
[0088] 017<Field name = " LastName " />
[0089] 018<Field name = " State " />
[0090] 019</Results>
[0091] 020</QueryAbstraction>
```

[0092] 图解性地,表 I 中所示的抽象查询包括选择规格(行 005-014),选择规格包含选择标准和结果规格(行 015-019)。在一个实施例中,选择标准由域名(对于逻辑域),比较操作符(=, >, <, 等等)和值表达式(正进行比较的域)组成。在一个实施例中,结果规格是要作为查询执行结果回送的抽象域的列表。抽象查询中的结果规格可以由域名和排序标准组成。

[0093] 以下在表 II 中示出了对应于表 I 中抽象查询的数据库抽象部件 148 的图解实例。通过图解方式,使用 XML 定义数据库抽象 148。然而,可以使用任何其他语言。

[0094] 表 II 数据库抽象示例

```
[0095] 001<? xml version = " 1.0 " ? >
[0096] 002<DataRepository>
[0097] 003<Category name = " Demographic " >
[0098] 004<Field queryable = " Yes " name = " FirstName " displayable
= " Yes " >
[0099] 005<AccessMethod>
[0100] 006<Simple columnName = " f_name " tableName = " contact " >/Simple>
[0101] 007</AccessMethod>008<Type baseType = " char " ></Type>
[0102] 009</Field>
[0103] 010<Field queryable = " Yes " name = " LastName " displayable = " Yes " >
[0104] 011<AccessMethod>012<Simple columnName = " l_name " tableName
= " contact " ></Simple>
[0105] 013</AccessMethod>
[0106] 014<Type baseType = " char " ></Type>
[0107] 015</Field>
[0108] 016<Field queryable = " Yes " name = " State " displayable = " Yes " >
[0109] 017<AccessMethod>018<Simple columnName = " state " tableName
= " contact " ></Simple>
[0110] 019</AccessMethod>
[0111] 020<TypebaseType = " char " ></Type>
[0112] 021</Field>
[0113] 022</Category>
[0114] 023</DataRepository>
[0115] 图 3 示出了图解性运行时方法 300,其示例了运行时部件 150 操作的一个实施例。
```

当运行时部件 150 接收抽象查询的实例作为输入时（例如，图 2 中所示的抽象查询 202），在步骤 302 进入方法 300。在步骤 304，运行时部件 150 读取并解析抽象查询的实例，并定位各选择标准和所希望的结果域。在步骤 306，运行时部件 150 进入循环（包括步骤 306，308，310 和 312），以处理抽象查询中出现的每个查询选择标准声明，从而构建具体查询的数据选择部分。在一个实施例中，选择标准由域名（对于逻辑域），比较操作符（=，>，<，等等）和值表达式（正进行比较的域）组成。在步骤 308，运行时部件 150 使用来自抽象查询的选择标准的域名在数据库抽象部件 148 中查找域的定义。如上所述，域定义包括用于存取与域相关的物理数据的存取方法的定义。之后，运行时部件 150 构造（步骤 310）正被处理的逻辑域的具体查询成分（querycontribution）。如在这里定义的，具体查询成分在具体查询的一部分，其被用于根据当前逻辑域执行数据选择。具体查询是用象 SQL 和 XML 语言表示的查询，并且与所给物理数据库（例如，关系数据库和 XML 数据库）的数据一致。因此，具体查询被用于定位和检索物理数据库（如图 1 中数据库 156-157 所示）的数据。然后，将针对当前域生成的具体查询成分添加到具体查询声明中。然后，方法 300 返回到步骤 306，以便开始处理抽象查询的下一个域。因此，对于抽象查询中的每个数据选择域，重复在步骤 306 进入的过程，从而将附加内容加到最终要执行的查询中。

[0116] 构造完具体查询的数据选择部分后，运行时部件 150 识别要作为查询执行结果返回的信息。如上述，在一个实施例中，抽象查询定义要作为查询执行结果返回的抽象域的列表，在这里被称为结果规格。抽象查询中的结果规格可以由域名和排序标准组成。因此，方法 300 在步骤 314 进入循环（由步骤 314，316，318 和 320 定义），以便将结果域定义加到正生成的具体查询中。在步骤 316，运行时部件 150 在数据库抽象部件 148 中查找结果域名（来自抽象查询的结果规格），然后从数据库抽象 148 检索结果域定义，以便识别当前逻辑结果域要返回的数据的物理位置。然后，运行时部件 150 构造（如步骤 318）逻辑结果域的（具体查询中标识要被返回的数据的物理位置的）具体查询成分。然后，在步骤 320，将具体查询成分加到具体查询声明中。一旦处理完抽象查询中的每一个结果规格，则在步骤 322 执行查询。

[0117] 参照图 4，根据步骤 310 和步骤 318，描述用于构造逻辑域的具体查询成分的方法 400 的一个实施例。在步骤 402，方法 400 查询与当前逻辑域相关的存取方法是否简单存取方法。如果是简单存取方法，根据物理数据位置信息构造具体查询成分（步骤 404），然后，根据上述方法 300，继续进行处理。否则，继续处理到步骤 406，以查询与当前逻辑域相关的存取方法是否过滤存取方法。如果是过滤存取方法，根据某物理数据实体的物理数据位置信息，构造具体查询成分（步骤 408）。在步骤 410，用附加逻辑（过滤器选择）扩展具体查询成分，该逻辑用于与物理数据实体相关的子集数据。然后，根据上述方法 300，继续进行处理。

[0118] 如果存取方法不是过滤存取方法，处理从步骤 406 进行到步骤 412，在步骤 412 中，方法 400 查询存取方法是否合成存取方法。如果存取方法是合成存取方法，在步骤 414 定位并检索合成域表达式中每个子域引用的物理数据位置。在步骤 416，合成域表达式的逻辑域引用被合成域表达式的物理位置信息替换，从而生成具体查询成分。然后，根据上述方法 300，继续进行处理。

[0119] 如果存取方法不是合成存取方法，处理从步骤 412 进行到步骤 418。步骤 418 代表

如本发明实施例所预期的任何其他存取方法类型。然而,应当理解,在其中所考虑的实施例比所有可实现的存取方法少。例如,在特定实施例中仅使用了简单存取方法。在另一个实施例中,只使用了简单存取方法和过滤存取方法。

[0120] 如上所述,如果逻辑域规定的格式不同于底层物理数据,必须执行数据转换。在一个实施例中,当根据方法 400 构造逻辑域的具体查询成分时,针对各自的每个存取方法执行初始转换。例如,可以作为步骤 404, 408 和 416 的一部分,或者随后立即执行转换。当在步骤 322 执行查询之后,执行从物理数据格式到逻辑域格式的转换。当然,如果逻辑域定义的格式与底层物理数据相同,则不必进行转换。

[0121] 数据库抽象部件的其他实施例

[0122] 在一个实施例中,为每个单独的物理数据表示 214 提供不同的单数据库抽象部件 148(在图 2B 和 2C 中)。在可选实施例中,单数据库抽象部件 148 包含用于两个或更多个物理数据表示 214 的域规格(具有相关存取方法)。在另一个实施例中,提供多个数据库抽象部件 148,其中每个数据库抽象部件 148 公开相同底层物理数据(其可以包括一个或多个物理数据表示 214)的不同部分。这样,多个用户可以同时使用单个应用 140 来存取同一底层数据,其中,由各自的数据库抽象部件 148 确定暴露给应用的底层数据的特定部分。在名称为“DYNAMIC ENDUSER SPECIFIC CUSTOMIZATION OF AN APPLICATION'S PHYSICAL DATA LAYER THROUGH A DATA REPOSITORY ABSTRACTION LAYER”(应用的物理数据层到数据库抽象层的动态终端用户特定定制),并转让给国际商用机器公司的待审美国专利申请(代理案卷 ROC920020088)中,更具体地说明了此后者实施例。

[0123] 在任何实例中,数据库抽象部件 148 包含(或引用)至少一个存取方法,该方法将逻辑域映射到物理数据。对于这一点,如前述实施例说明的,存取方法描述了定位并操作对应于逻辑域的数据的物理表示的手段。

[0124] 在一个实施例中,扩展数据库抽象部件 148 到包括大量数据源的描述,数据源可以是本地的和/或分布于整个网络环境。数据源可以使用大量不同数据表示和数据存取技术。在一个实施例中,除了用于存取数据的方法以外,这通过用定义与逻辑域相关的数据的位置的位置规格配置数据库抽象部件 148 的存取方法来实现。

[0125] 现在参照图 5,所示为具有多个数据源(库)502 的环境 500 的逻辑/运行时视图,并图解了在这样的环境中数据库抽象部件 148 的操作的一个实施例。要经由数据库抽象部件 148 存取的数据源 502 可以是本地的,远程的或者两者都有。在一个实施例中,数据源 502 代表图 1 中所示的数据库 156-157。通常,数据库抽象部件 148 与上述那些实施例配置相似。同样地,数据库抽象部件 148 有逻辑域定义和针对每个逻辑域定义的相关存取方法。然而,与其他只存取单一数据源的实施例比较,除了物理表示规格之外,现在还用位置规格配置存取方法。位置规格描述了位置(即,数据源),在位置规格中定位要存取的数据(即,与逻辑域定义相关的数据)。然而,在一个实施例中,可以预期,某些存取方法可以不用位置规格进行配置,而是默认的指向本地数据源。

[0126] 通常,图 5 示出了应用 140,抽象查询规格 142(这里也被称为应用查询规格),数据库抽象部件 148(用于将逻辑域映射到存取方法),和运行时部件 150,运行时部件 150 负责将抽象查询转变成数据库 502 所支持的一个或多个数据存取请求,该数据库 502 包含正被查询的物理信息。与以上描述的一些实施例相比较,图 5 的数据库抽象部件 148 和运行

时部件 150 被配置为支持具有相关数据的逻辑域的定义和查询,所述相关数据可以分布于多个本地和 / 或远程物理数据库 502(在这里也被称为本地 / 远程数据源 502) 上,并且可以经由大量基于查询和基于过程的接口进行存取。

[0127] 针对这一点,应用 140 根据抽象查询规格 142 定义其数据需求,抽象查询规格 142 包含查询选择和 / 或基于逻辑域,而不涉及实际数据的物理位置或表示的更新逻辑。数据库抽象部件 148 包括逻辑域定义 504 和针对每个逻辑域的存取方法 506。逻辑域定义 504 描述应用 140 可用的逻辑域。在一个方面,数据库抽象部件 148 管理应用 140 可以使用的信息。从而使得应用能使用在新的本地或者远程数据源中新逻辑域的增加。每个存取方法 506 定义了逻辑域和本地 / 远程数据源 502 中逻辑域的物理表示之间的映射。可以参照图 6 理解此关系。

[0128] 图 6 图解示出了抽象查询 602,抽象查询 602 包括多个逻辑域 $604_1, \dots, 604_n$ (合称为逻辑域 604)。通过定义特定数据库抽象部件 148,每个逻辑域 604 与存取方法 $608_1, \dots, 608_n$ (全体称为存取方法 608) 相关联(线 606 表示)。存取方法 608 中的物理表示信息包括要使用的存取方法的名称(这里用“F1 的存取方法”,“F2 的存取方法”等等表示),和要传递给所命名的存取方法、描述如何存取与逻辑域相关的物理数据的多个参数。通常,这样的参数包括定位符参数 $610_1, \dots, 610_n$ (合称为定位符参数 610;在这里还被称为位置规格)和其他存取数据所需的存取参数。所给数据库抽象部件实例可以表示由多个本地和远程物理数据库管理的信息。

[0129] 在图 7-8 中示出了图解实施例,其中可以用位置规格和其他存取数据所需的存取参数配置数据库抽象部件实例。首先参照图 7,所示为用关系存取方法配置的数据库抽象部件的域规格 700。域规格 700 特别针对由域名称 702“CreditRatingDescription”标识的特定逻辑域,并具有所关联的存取方法。所关联的存取方法名称 704“simple-remote”指出存取方法是简单域存取方法,其中逻辑域被直接映射到底层物理数据表示的特定实体,并且数据是远程定位的。在此实例中,逻辑域被映射到给定数据库表“credit_t”和列“desc”。“URL”是位置规格(定位符参数),其规定了物理数据的位置。在此实例中,“URL”包括使用的 JDBC 驱动程序的标识符,存有数据的远程系统名称(remotesystem.abc.com)和包含数据的数据库模式(creditscheme)。“JDBC Driver”是针对此类远程数据库实施 SQL 存取的 JAVA 类的名称。

[0130] 现在参照图 8,所示为用过程存取方法配置的数据库抽象部件的域规格 800。域规格 800 特别针对由域名称 802“creditRating”标识的特定逻辑域,并具有所关联的存取方法。所关联的存取方法名称 804“procedural”指出存取方法是过程存取方法。“Service Spec”标识要存取的 web 服务的 web 服务描述语言(WSDL)定义。WSDL 是 web 服务的标准接口定义语言。web 服务是用于启动软件应用的标准方法,该软件应用使用已建立的 web 通信基础设施和标准的数据表示技术,例如 XML 来表示调用方应用和所启动的 web 服务之间传递的信息。“Service Name”标识“Service Spec”内定义的可能服务组中要被访问的 web 服务的名称。“Port Name”标识“Service Spec”内定义的可能端口名称组中要访问的服务的端口名称。所命名的端口定义服务的网络位置。“Operation”是要启动的操作的名称。web 服务能支持多于一个的被称为“操作”的函数。当启动 web 服务时,“Input”标识所需的输入。在此实例中,提供了 last name 值作为对服务的输入。“Output”标识与此逻辑

辑域相关的输出数据项。当服务被调用时,服务可返回几部分输出。因此,“Output”标识定义与当前逻辑域相关的该部分输出数据。

[0131] 注意,在过程存取方法实例中,本地数据的数据库抽象部件的域规格看起来与图 8 中所示的用于存取远程数据的域规格 800 基本相似。唯一的区别在于,在本地实例中,引用的 WSDL 文档会有回指向服务正在运行的本地服务器的 URL。

[0132] 再次参照图 5,现在描述运行时部件 150 操作的一个实施例。通常,运行时部件负责根据抽象查询构造并执行可执行的查询。针对此点,在模块 510,运行时部件 150 解析抽象查询,并使用数据库抽象部件 148 将针对一个或多个逻辑域的引用映射到其相应的物理位置和存取方法(在这里,全体被合称为存取方法 506)。在一个实施例中,运行时部件 150 划分(块 512)全部物理数据查询需求为组(被称为“子查询”514),组表示使用相同存取方法存取同一物理源。然后执行“子查询”(块 516)。在全部查询结果 520 被回送给应用 140 之前,合并并规格化每一个子查询 514 的结果(块 518)。在一方面,此查询划分实现方案容许运行时部件 150 利用多处理器硬件架构并行地运行多个子查询。

[0133] 在一个实施例中,运行时部件 150 还管理本地数据高速缓存 522。本地数据高速缓存 522 包含针对某些逻辑域检索的数据,并且在随后查询期间被用作查找允许高速缓存的数据库抽象部件中所标识的逻辑域的第一选择。以被高速缓存的形式进行有利地管理的逻辑域是那些取值相对静态并且/或者导致显著存取开销的逻辑域(在这里,按照获取数据所需的时间,或存取数据的费用开支来测量开销,假定在每次使用付费(pay-per-use)模型中管理某些信息)。

[0134] 在各种实施例中,优选提供了超出现有技术的许多优点。在一个方面,优选通过在应用查询规格和底层数据表示之间定义松耦合来获得优势。不是用特定的表,列和关系信息对应用编码,如使用 SQL 的实例,应用以更抽象的形式定义数据查询需求,之后在运行时,将该需求绑定到特定物理数据表示。即使修改了底层数据表示,或者请求方实体将被用于的物理数据表示比开发请求方实体时使用的物理数据表示更新,松查询数据耦合仍然优选地允许请求方实体(例如,应用)正常工作。在修改或重新构造所给物理数据表示的实例中,优选地更新相应数据库抽象以反映对底层物理数据模型的改变。查询可使用相同的逻辑域组,并且该域组仅仅被绑定到物理数据模型中的不同实体或位置。结果,即使相应的物理数据模型已经经过重大的改变,写到抽象查询接口的请求方实体仍然继续工作,并且没有改变。在请求方实体将用于比开发请求方实体时所使用的更新的物理数据表示的情况下,可以使用相同的技术(例如,关系数据库),但遵循不同的命名和组织信息策略(例如,不同的模式),以实现新的物理数据模型。新模式将包含可以通过使用简单,过滤以及合成域存取方法技术被映射到应用所需的逻辑域组的信息。可选地,新的物理表示可以使用可选的技术来表示相似的信息(例如,相对关系数据库系统,使用基于 XML 的数据库)。在任一情况下,通过提供可选数据库抽象(其将查询中引用的域与新物理数据模型中的位置和物理表示相映射),被编写为使用抽象查询接口的请求方实体能轻易地移植为使用新物理数据表示。

[0135] 在另一个方面,易于使用应用构造器,并且便利终端用户。在底层物理数据库中,使用抽象层表示逻辑域,使应用开发者能关注于关键应用数据需求,而不必关心底层数据表示的细节。结果,在应用开发过程中获得更高的产出,并减少了错误率。针对终端用户,数

数据库抽象优选提供数据过滤机制,公开相关数据并隐藏开发指定查询的特定类别终端用户不需要的无关内容。

[0136] 此外,能够有利地利用存在多个数据源的情况。通过用位置规格配置数据库抽象部件,能更适宜地存取多个数据源,不管数据源是本地的或是远程的。以此方式,提供了能够利用当今流行的分布式环境的基础结构。

[0137] 实现此模型的解决方案使用所提供的抽象查询规格来描述其信息需求,而不用考虑所涉及数据的位置或者表示。查询被提交给运行时部件,运行时部件使用数据库抽象部件确定用于存取查询中表示的每个信息逻辑片的位置和方法。在一个实施例中,运行时部件还包括上述数据高速缓存功能,以便访问数据高速缓存。

[0138] 优选地,此模型容许以独立于解决方案所使用的数据的物理位置或表示的方式开发解决方案,使得能将解决方案容易地部署到许多不同数据拓扑上,并且在随时间推移而重新定位或重新组织数据的情况下,允许解决方案仍然有效。优选地,此解决方案还简化了扩展方案以利用附加信息的任务。在抽象查询级实现扩展,并且不需要增加特定于所存取的新数据的位置或表示的软件。此方法优选地为软件应用提供独立于用于存取数据的特定方法和所引用的每个数据项的位置的通用数据存取方法。可以通过关系模型(在现有的关系数据库系统中),层次模型(如 XML)或者某些其他物理数据表示模型表示经由抽象查询所存取的物理数据。还支持大量数据存取方法,包括那些基于现有的数据查询方法,例如 SQL 和 XQuery 的方法,以及涉及编程访问信息的方法,例如通过 web 服务调用(例如使用 SOAP)或通过 HTTP 请求检索数据。

[0139] 应当注意到,在这里涉及的任何特定值,定义,程序语言和示例,仅用于说明目的。因此,本发明不限于任何特定的说明和示例。另外,虽然参参考选择操作说明了本发明的各个方面,然而其他输入/输出操作也优选得到考虑,包括熟知的操作,例如,增加,修改,插入,删除,等等。当然,某些存取方法会约束抽象查询函数的类型,其可以使用利用了特定存取方法的域进行定义。例如,涉及合成存取方法的域不是修改,插入和删除的可行目标。

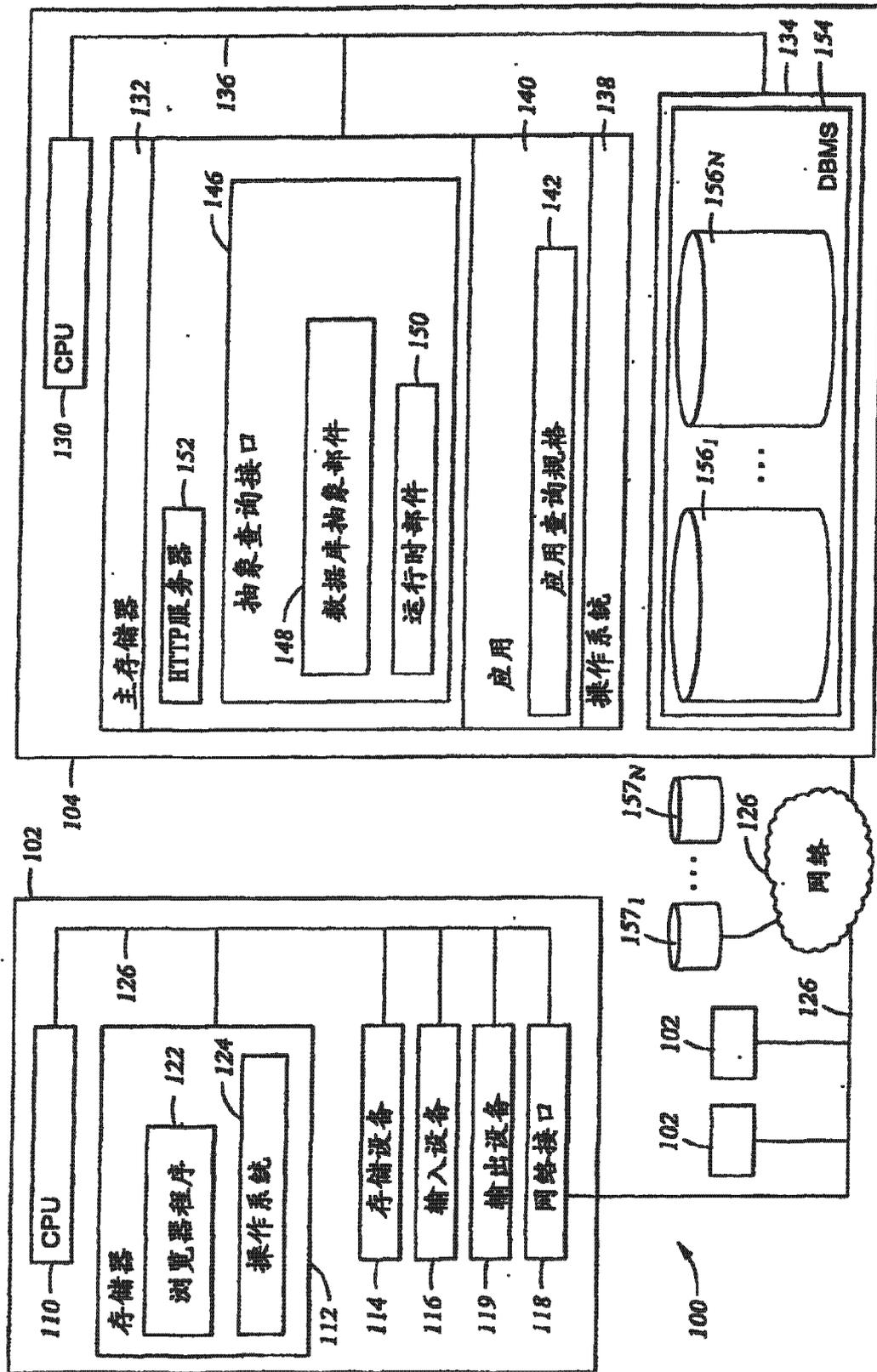


图 1

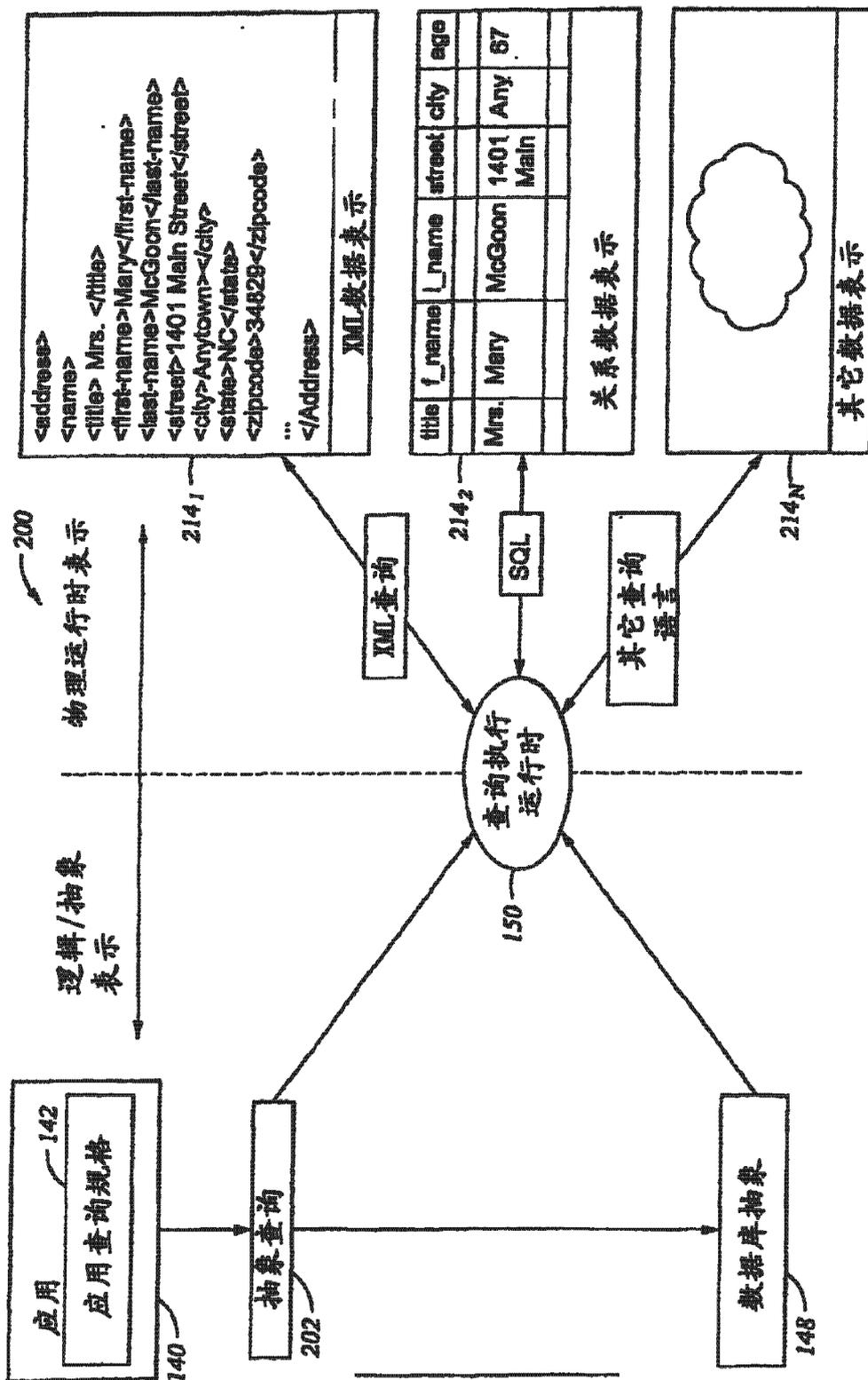


图 2A

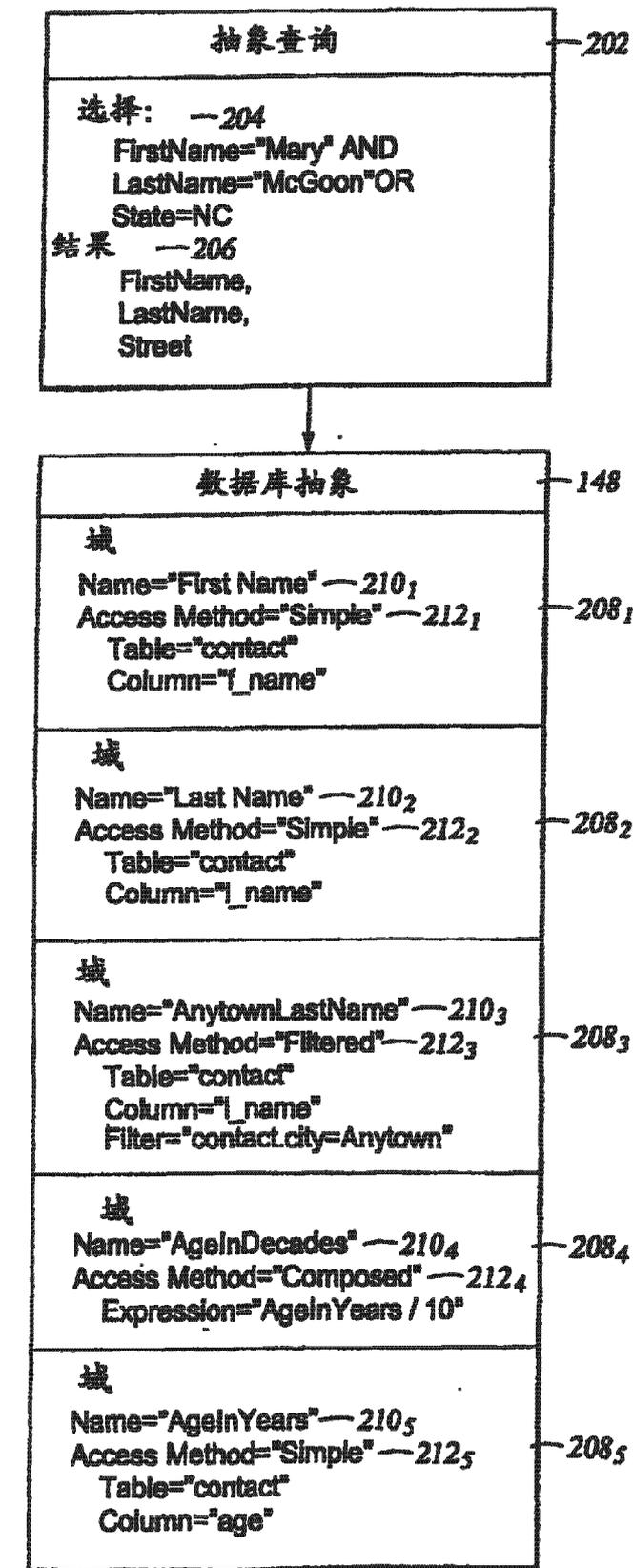


图 2B

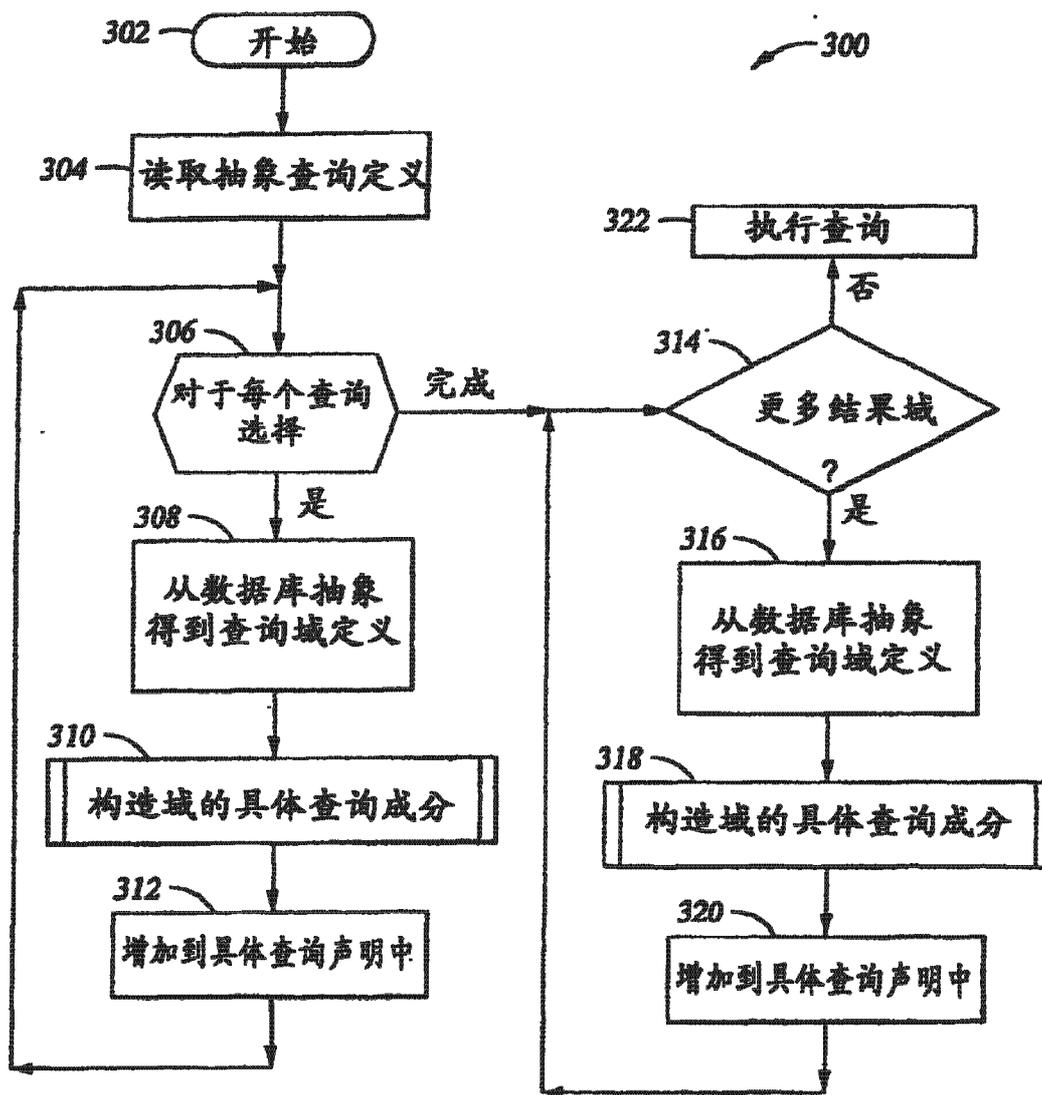


图 3

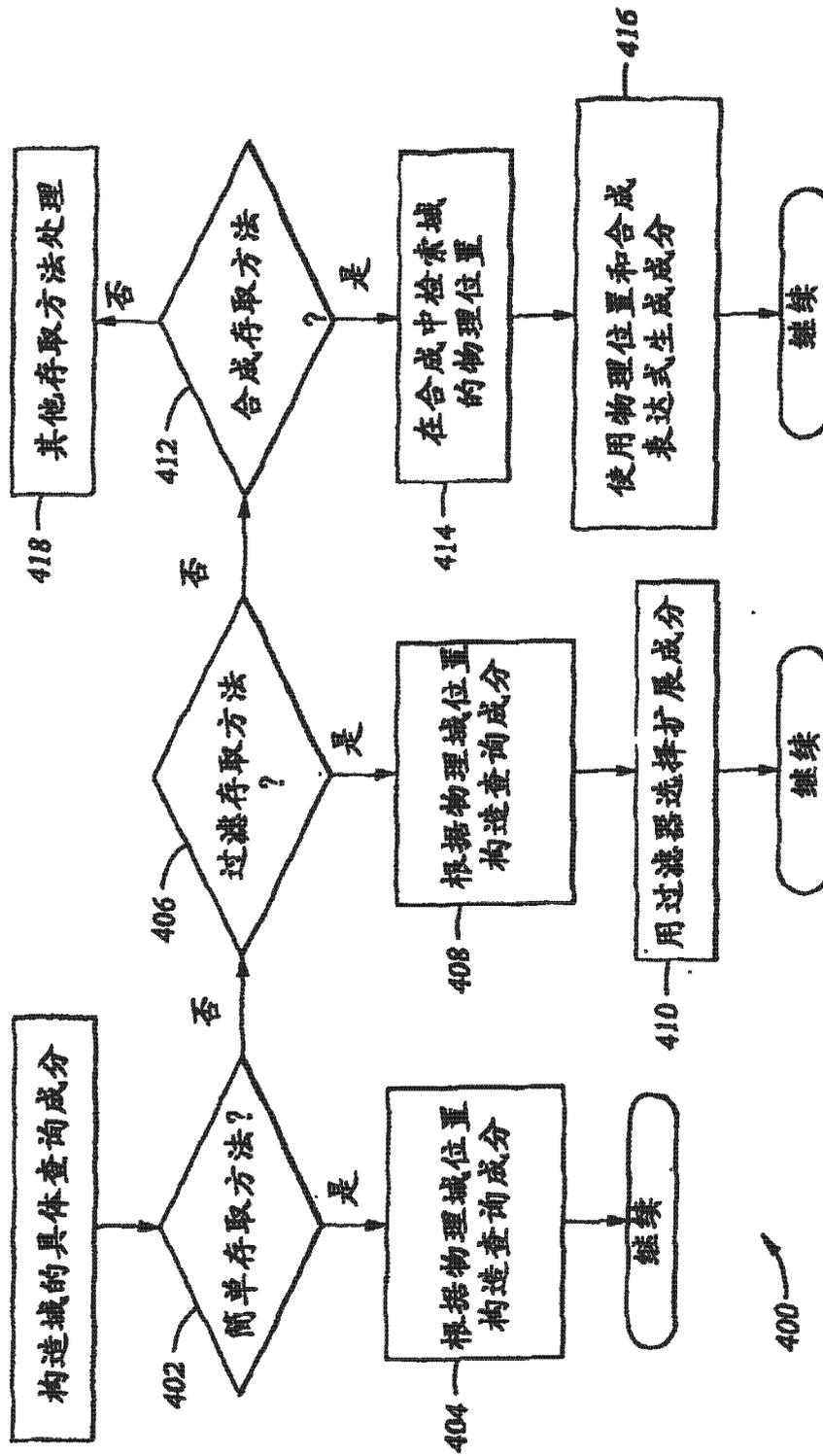


图 4

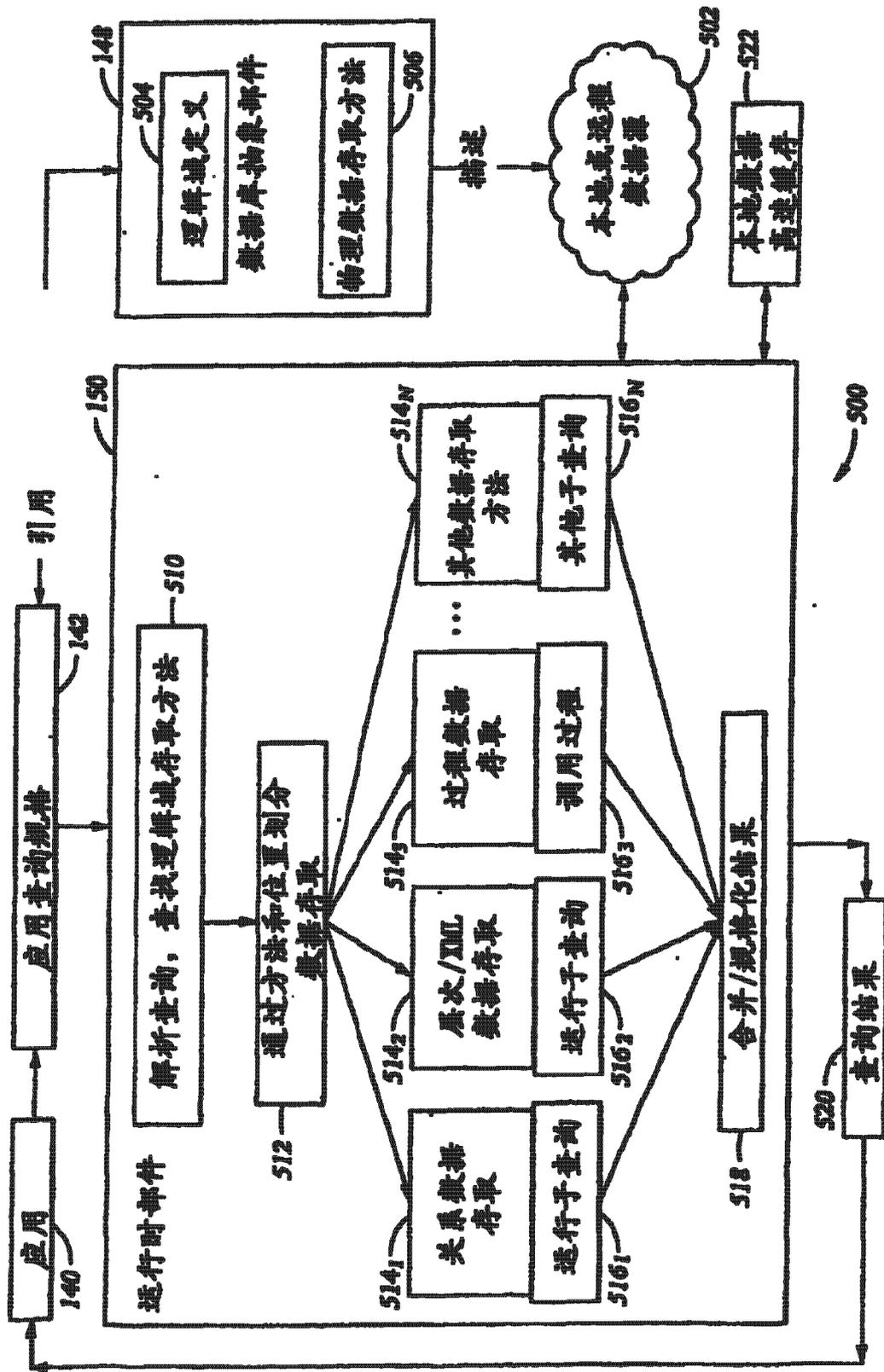


图 5

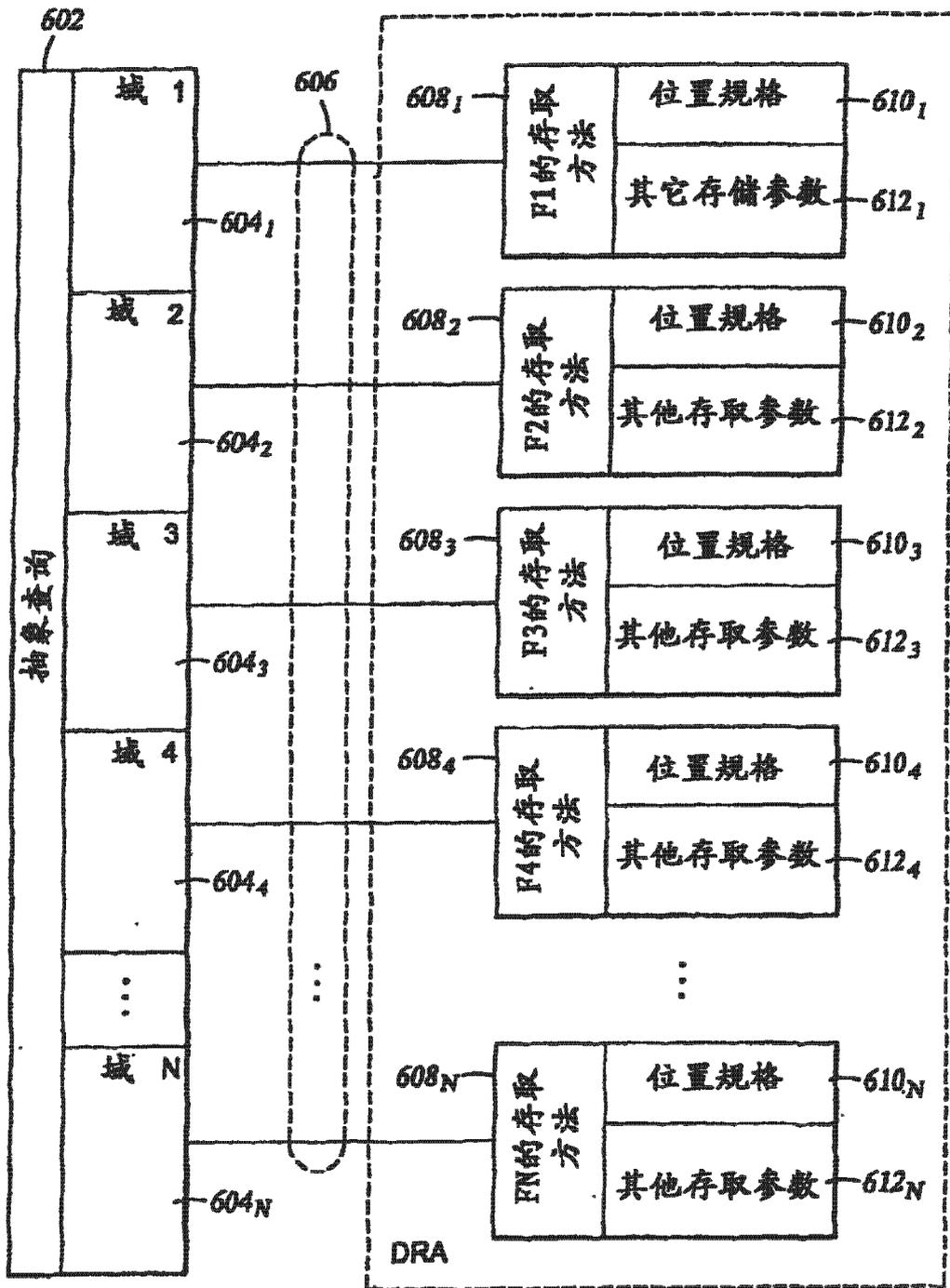


图 6

700

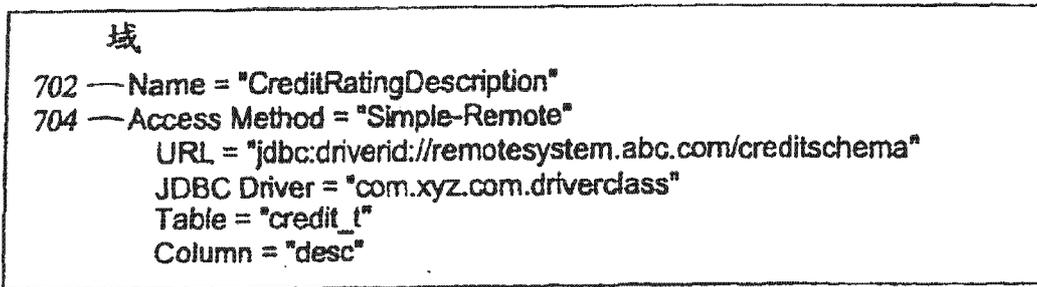


图 7

800

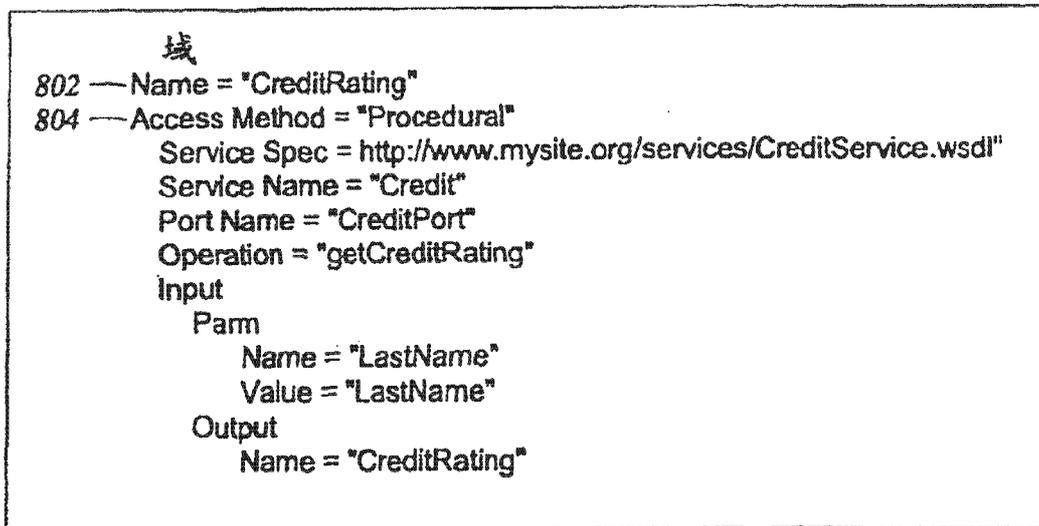


图 8