

[19] 中华人民共和国国家知识产权局

[51] Int. Cl.
G06F 17/30 (2006.01)



[12] 发明专利说明书

专利号 ZL 03804980.5

[45] 授权公告日 2009年3月18日

[11] 授权公告号 CN 100470538C

[22] 申请日 2003.2.14 [21] 申请号 03804980.5
[30] 优先权
[32] 2002.3.5 [33] US [31] 10/091,919
[86] 国际申请 PCT/GB2003/000679 2003.2.14
[87] 国际公布 WO2003/075176 英 2003.9.12
[85] 进入国家阶段日期 2004.8.31
[73] 专利权人 国际商业机器公司
地址 美国纽约
[72] 发明人 兰德尔·詹姆斯·理查德 蔡文宪
胡大伟 张立莉
肯尼思·卡林·纳尔逊
[56] 参考文献
WO0175679 A1 2001.10.11
US5913205 A 1999.6.15
US6078926 A 2000.6.20

US5724575 A 1998.3.3
WO0045294 A1 2000.8.3
WO0201384 A2 2002.1.3

审查员 李宁

[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 付建军

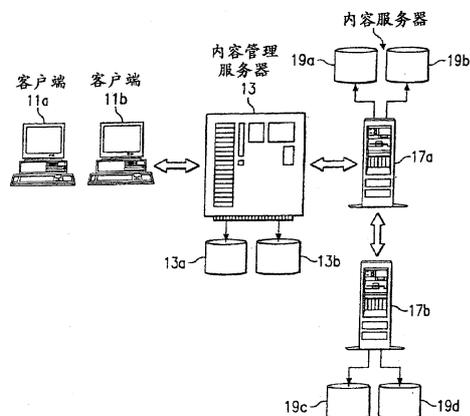
权利要求书 2 页 说明书 10 页 附图 4 页

[54] 发明名称

对多个内容管理数据模型的支持

[57] 摘要

用于管理多媒体内容的方法、系统和程序产品。它们以特定的多媒体内容类型的高级别的内容模型和用于映射到数据引擎的低级别的物理模型的组合作基础。所述方法、系统和程序产品包括表示多媒体内容管理对象和管理关系数据库中的对象。关系数据库用于以多个关系数据库表来表示数据，而每一个关系数据库表具有带有多个列的至少一行。对内容是这样进行管理的：将对象作为项目关联到第一关系数据库表中的行；并如果有对象的另外的组件的话，将其关联到另外的关系数据库表。



1. 一种将多媒体内容管理对象表示为关系数据库中的项目的方法，所述关系数据库适用于以一个或多个关系数据库表表示多媒体内容管理数据，每一个所述关系数据库表都具有带有多个列的至少一行，所述方法包括：

将内容管理对象的根组件关联到第一关系数据库表中的一行；

将根组件的属性关联到第一关系数据库表的对应的列；

如果有内容管理对象的另外的组件的话，将其关联到另外的关系数据库表中的行；以及

使用关系数据库表中的项目作为构件块构建多个高级别的内容管理数据模型，每个内容管理数据模型对应于不同的应用。

2. 根据权利要求 1 所述的方法，其中，每一个另外的组件包括根组件的子组件或根组件之外的另一个组件的子组件。

3. 根据权利要求 2 的方法，使用子组件中的外部键来引用其父组件。

4. 根据权利要求 1 所述的方法，其中，属性包括指向单独的仓库中存储的数据源的指针。

5. 根据权利要求 1 的方法，其中属性包括指向另一个内容管理对象的指针。

6. 根据权利要求 1 所述的方法，其中，所述一个或多个关系数据库表中的行包括源项目和目标项目之间的链接。

7. 一种系统，用于管理多媒体内容管理对象的一个或多个多媒体数据对象项目并将多媒体数据对象项目从多媒体数据对象内容仓库通过多媒体数据对象内容服务器提供到客户端，其中，每一个多媒体数据对象项目都包括多媒体数据对象属性和组件，并且其中，多媒体数据对象内容服务器被控制并被配置为：

a. 将内容管理对象的根组件关联到第一关系数据库表中的一行；

- b. 将根组件的属性关联到第一关系数据库表的对应的列；以及
- c. 如果有内容管理对象的另外的组件的话，将其关联到另外的关系数据库表中的对应的行；

使用数据对象项目构建多个高级别的内容管理数据模型，每个内容管理数据模型对应于不同的应用。

8. 一种将多媒体内容管理对象表示为关系数据库中的项目的设备，所述关系数据库适用于以一个或多个关系数据库表表示多媒体内容管理数据，每一个所述关系数据库表都具有带有多个列的至少一行，所述设备包括：

用于将内容管理对象的根组件关联到第一关系数据库表中的一行的装置；

用于将根组件的属性关联到第一关系数据库表的对应的列的装置；

用于如果有内容管理对象的另外的组件的话，将其关联到另外的关系数据库表中的行的装置；以及

用于使用关系数据库表中的项目作为构件块构建多个高级别的内容管理数据模型的装置，每个内容管理数据模型对应于不同的应用。

9. 根据权利要求 8 所述的装置，其中，每一个另外的组件包括根组件的子组件或根组件之外的另一个组件的子组件。

10. 根据权利要求 9 的装置，使用子组件中的外部键来引用其父组件。

11. 根据权利要求 8 所述的装置，其中，属性包括指向单独的仓库中存储的数据源的指针。

12. 根据权利要求 8 的装置，其中属性包括指向另一个内容管理对象的指针。

13. 根据权利要求 8 所述的装置，其中，所述一个或多个关系数据库表中的行包括源项目和目标项目之间的链接。

对多个内容管理数据模型的支持

技术领域

本发明涉及数据结构模型，包括多媒体数据结构模型，用于促进内容管理应用程序的访问、检索和数据转换。数据结构模型用于组织、关联和访问数据和文件，包括关系、网络、分层的和实体关系模型，以及图形、文字和 Web 页文件等等。

背景技术

多媒体数据管理和传送是一个复杂的消耗计算机和内存的任务。这里所使用的“多媒体”数据是指声音、图形、动画、文字和视频数据的组合，它们可能在存储和检索的关联的系统中链接起来，可能是交互式的，并可能链接到其他媒体。

多媒体内容管理系统需要管理大量的不同的数据，包括非结构化数据以及结构化数据。结构化数据常常包括描述非结构化数据的“元数据”，以便可以以各种方式对后者进行组织、分组和关联，搜索、检索和管理。在内容管理系统内，常常使用关系数据库管理系统 (RDBMS) 管理结构化数据，以便可以使用现成的 RDBMS 数据管理技术。然而，非结构化数据的类型，即，多媒体内容，在不同的应用程序之间各有不同。

由于数据的类型（包括其架构、元数据、适用的扩展程序等等）在不同的应用程序和内容数据模型（如，流式、图形、可视、视频、音频、文字、数字）之间各不相同，因此，需要不同的数据库设计来支持每一个应用程序和数据模型的元数据和架构数据库。迄今为止，未能以一个单一的、通用内容管理系统支持使用不同内容数据模型的不同内容管理应用程序。这样的系统不能具有一个预先设计的，即，“万能的”数据库管理系统。相反，必须给内容管理系统配备允许应用

程序开发人员描述所需的结构化数据的功能，以便可以为该应用程序创建具有适当的设计的数据库。此描述通常以内容管理系统支持的高级别的数据模型的抽象形式来表达，通常通过应用程序编程接口 (API) 或图形用户界面 (GUI) 进行。尽管这种高级别的数据模型提供支持一个以上的应用程序的灵活性，但是，它也仅限制于可以以该特定高级别模型表达的那些应用程序。

数据模型是内容管理系统的基本的方面，因为它确定了系统的潜在功能以及其局限性。然而，尽管数据模型有时可以在某种程度上得到扩展，但是，这样的扩展通常是非常有限的，并不能捕获不同的范例，特别是具有不兼容的甚至相互冲突的概念的范例。

如此，很显然，需要一种可扩展并且可缩放的数据模型，该数据模型应该能够容易地支持各种各样的内容数据模型，并可以以系统或程序产品的方式实现。

还需要工具、工具包、工具箱或向导为可扩展并且可缩放的内容管理 RDBMS 创建和填充不同内容架构和元数据的关系数据库。

发明内容

相应地，本发明提供了表示数据库中的多媒体内容管理对象的方法。根据本发明的方法，使用了数据库，所述数据库包括多媒体内容数据的一个或多个高级别的内容模型以及低级别的物理模型。低级别的物理模型提供到数据引擎的映射。该方法包括以低级别的物理表示输入多媒体数据元数据和架构，以及将元数据和架构映射到数据引擎的步骤。优选情况下，数据引擎是关系数据库管理系统、面向对象的数据库管理系统或 XML 数据仓库。

优选情况下，该方法提供到数据引擎的映射，所述映射进一步包括将多媒体内容管理对象表示为关系数据库中的项目的方法，所述关系数据库适用于以一个或多个关系数据库表表示多媒体内容管理数据。每一个关系数据库表都具有带有多个列的至少一行。该方法包括下列步骤：将对象的根组件作为一个项目关联到第一关系数据库表中

的一行；将根组件的属性关联到第一关系数据库表的对应的列；如果有对象的另外的组件的话，将其关联到另外的关系数据库表中的行；以及使用项目作为构件块以构建多个高级别的内容管理数据模型。

相应地，本发明进一步提供一种包括指令的计算机程序产品，当在具有非易失性存储设备的数据处理系统上执行所述指令时，使所述系统执行如上所述的方法。

相应地，本发明进一步提供一种系统，用于管理多媒体数据对象项目并将多媒体数据对象项目从多媒体数据对象内容仓库通过多媒体数据对象内容服务器提供到客户端。多媒体数据对象包括多媒体数据对象项目，每一个多媒体数据对象项目都包括多媒体数据对象属性和组件。多媒体数据对象内容服务器被控制并被配置为将对象的根组件关联到第一关系数据库表中的一行；将根组件的属性关联到第一关系数据库表的对应的列；并如果有对象的另外的组件的话，将其关联到另外的关系数据库表中的对应的行。所述项目被用作构件块以构建多个高级别的数据模型。

附图说明

现在将参考下面的图形，只作为示例，对本发明的优选实施例进行详细的描述：

图 1 说明了具有多个客户端、内容管理服务器，以及多个内容服务器的多媒体数据库系统。

图 2 说明了本发明的方法、系统和程序产品的分层结构，以 API 的结构作为高级别的数据模型，以通用、低级别的基础结构作为物理数据模型，以及基础的关系数据库管理系统。

图 3 说明了图 2 的关系数据库管理系统的简化的总览，显示了分层的 RDBMS 表，包括具有根组件的行的项目表，并包括组件和属性单元格（列），具有子组件和属性单元格（列）的组件表，子组件和属性的孙子表，并带有适当的指针。

图 4 描述了图 3 的 RDBMS 表之间的关系，并说明了项目类

型的并排的映射，包括映射到项目的根表、子表，以及孙子表，作为项目类型的实例，包括根组件、子组件和孙子组件。每一个组件都是对应的表中的行。

图 5 说明了图 2 的 RDBMS 的表中行，关联源项目、目标项目以及链接项目。

图 6 说明了将资源引用应用于目标项目。

图 7 说明了将多媒体内容存储在各种服务器上以便在终端上（例如在安全系统中）使用的个人标识应用程序。

具体实施方式

如图 1 所示，多媒体系统是复杂的客户端-服务器系统，具有多个客户端 11a、11b，服务器 13，以及一个或多个内容服务器 17a、17b，其中每一台服务器都具有一个或多个内容存储区 19a、19b、19c、19d。值得注意的是，内容存储区可以包含结构化数据，如内容的元数据，或非结构化数据，如文本数据、图像数据、流式音频或流式视频。这些数据可以采用并且通常采用各种不同的格式。

服务器 13 可以包括 Web 服务器，也包括内容管理服务器。内容管理服务器是这里所描述的内容管理 RDBMS 的仓库。尽管是就关系数据库管理系统对本发明进行说明的，但是也可以使用诸如面向对象的数据库管理系统之类的其他数据仓库或 XML 数据仓库。

为在高级别的数据模型中容纳一个以上的内容数据模型，可以使用可扩展的内容管理系统来引入适当的元数据和架构以及新 API，以支持新的高级别的数据模型。此扩展性由这样的体系结构来实现：所述体系结构围绕支持物理数据模型的低级别的基础结构而建立，并通过 RDBMS 表的层次结构来实现，RDBMS 表中包含内容和内容元数据作为项目、组件、子组件和属性，在行内还有对应的行和列（属性）。体系结构和关联的基础结构和物理数据模型被用作构件块以支持高级别的数据模型。此物理数据模型独立于应用程序特定的语义，直接并有效地映射到被设计成利用 RDBMS 功能的关系数据库，并

可以随着时间的推移加以丰富以扩展其功能和/或利用新的 RDBMS 功能。

此物理数据模型的设计点不同于应用程序开发人员所知道的高级别的数据模型的设计点。高级别的数据模型主要侧重于其逻辑方面（其相对于一类应用程序的可用性和可表达性），而物理数据模型主要侧重于其物理方面（存储访问效率和利用 RDBMS 功能的能力），以提供对多个内容管理应用程序都有广泛用途的通用的基本功能。此物理数据模型，或它的任何一部分，可以也可以不通过高级别的数据模型暴露给用户。

在概念上，如图 2 所示，本发明的方法、系统和程序产品利用能够支持内容应用程序要求的不同而开放的集合以及多个高级别的数据模型，以及可扩展以接纳新技术、新标准和新要求的数据内容的低级别的物理模型，并提供到数据引擎（数据库管理系统作为关系数据库管理系统或面向对象的数据库管理系统，作为示例，而不作出限制）的有效映射。值得注意的是，每一个高级别的数据模型都呈现实现数据结构和限制的表示的应用程序编程接口的形式。还要注意的，低级别的数据模型或基础结构 25 可以支持多个高级别的模型 21、22、23。本发明的可扩展并且可缩放的内容管理系统包含一个或多个支持不同的高级别的数据模型 21、22、23 的 API，所述高级别的数据模型 21、22、23 又受通用、低级别的基础结构 25（即，物理数据模型）的支持，而低级别的基础结构 25 又受关系数据库管理系统 (RDBMS) 27 的支持。

图 3、4、5 和 6 说明了在 RDBMS 环境中物理模型和其表示之间的关系。在关系数据库中，数据是作为表来表示的。关系数据库表的构件块是行和列（属性）的表。具体来说，表由一行列标题连同零行或多行数据值构成。对于一个给定表 (1) 列标题指定一个或多个列，(2) 对于列标题行中指定的每一列，每一数据行正好包含一个值（或空值）。

由本发明的内容管理系统的物理层进行管理的内容的主要独立

的单元是“项目”。图 3 显示了项目数据库 30，具有根组件 31，具有项目行 32a 和 32b。所述项目包括组件 35 的带有类型的层次结构，一个或多个级别的子组件 38a、38b，具有属性 33a、33b、33c、33d、34a、34b、34c，并具有唯一标识符“ItemID”。

每一个组件 35 都包含对父组件的引用 36a。引用是通过外键进行的。

子组件表 38a 和 38b 可以引用其他的子组件或属性。子组件表 38b 是作为指向内容数据仓库 39a 和 39b 的引用指针来说明的。

图 4 是图 3 所示的模型的更一般、更高级别的视图。通过高级别的数据模型，或在应用程序内，“项目”，即，多媒体内容管理对象项目，可以用来将真实对象表示为文档、文件夹、媒体资产或其他真实世界的内容实体。图 4 说明了分层的物理项目及其属性 41b（在右边）到分层的 RDBMS 表示，即，RDBMS 项目类型 41a（在左边）的映射。“ItemType”是用户定义的架构和 RDBMS 表的命名的层次结构，其中每一个表都叫做“ItemType”的“ComponentType”²⁵。每一个“ComponentType”都包含用户指定的属性以及某些系统属性。

用户指定的属性可以是由 RDBMS 支持的任何数据模型，也可以是由内容管理系统支持的特殊数据模型。“项目”在逻辑上在相应的“ItemType”的根表中包括一行，在“ItemType”的每一子表中包括任意数量的行，条件是，“Item”中包含的子行的父行也必须包含在“项目”中。“项目”是存储在表的层次结构中的行的层次结构。这些行中的每一行都叫做“项目”的“组件”，并具有唯一地标识相应的“ComponentType”内“组件”的“ComponentID”。

可以为每一个“ComponentType”定义基数限制，以确保在它们的每一个父行下的该类型的最小行数和/或最大行数。默认最小行数是零，默认最大行数是无穷大。“ItemType”可以是有序的，也可以是无序的。对于有序的“ItemType”的“项目”，每一个父行下的子行都是有序的，顺序是由系统维护的。

“ItemType”可以带有版本号。如果是这样，则可以为“ItemType”

中的每一个“项目”维护“项目”的多个版本。它们共享同一个“ItemID”，但每一个版本都具有不同的版本号。可以为每一个带版本号的“ItemType”指定要为“项目”保留的最新的版本的数量。默认值是无穷大，即，系统将不自动对旧版本进行任何清除操作。

在一个优选实施例中，在一个“ItemType”内的两个“ComponentTypes”之间由 RDBMS 外键维护了父子关系，所述 RDBMS 外键是在引用父“ComponentType”的子“ComponentType”（或子组件）中创建的。如此，由 RDBMS 确保了每一个子“组件”（或子组件）到其父“组件”的引用完整性。在此实施例中，“项目”中的每一个“组件”都至少包含下列系统属性：“ItemID”、“ComponentID”、“ParentID”，可选地，还可以包含“项目”VersionNumber，以及 ACL 代码（访问控制列表代码）。虽然只有根“组件”需要“ItemID”、“项目”VersionNumber，以及 ACL 代码，但是它们也可以包括在每一个“组件”中，以促进访问。也可以包括其他系统属性以促进访问或维护，例如“ComponentType”、“组件”下面的直接的子组件的数量等等。还可以给为“ItemType”所定义的每一个“ComponentType”创建单独的 RDBMS 表，即，表不在两个“ItemType”之间共享。这种设计简化了逻辑，还增强了数据库可缩放性。在不同的实施例中，可以在多个“ItemType”之间共享表，以存储这些多个“ItemType”的定义类似的 ComponentType 的组件（行）。在此情况下，在此共享表中包括“ItemType”作为系统属性可以促进访问。当搜索范围涵盖包含正在被搜索的共同属性的多个“ItemType”时，以这种方式共享表可使搜索变得更快。

如图 5 所示，项目（包括子组件、孙子组件和属性）可以使用链接通过系统表彼此关联。链接是将源“Item”51 绑定到目标“Item”53，以及 LinkType 55 和可选链接“Item”57 的“外部”关系。这种关系是“外部的”，因为它没有嵌入在源“Item”或目标“Item”中。LinkType 是分类标签，以便应用程序可以有选择地对特定类型的链接进行遍历或搜索。链接“Item”是第三种“Item”，在必要时，可用于

描述这种关系。链接可以用来支持一对多和多对多关系，包括聚合和包含关系。

每一种关系的语义都由高级别的数据模型进行定义。一个示例是“foldering”，在这样的情况下，表示文件夹的“Item”链接到“包含”在此文件夹中的项目（表示文档或其他文件夹）。例如，源项目 51 充当目标项目 53 的容器。链接到源“Item”或目标“Item”的绑定可以是版本特定的，也可以是独立于版本的。前者绑定到项目的特定版本，而后者绑定到“Item”的任何/所有版本。在一个实施例中，链接是作为一个或多个专用 RDBMS 表中的行来维护的，所述表中至少包含下面的属性：源“ItemID”、源“Item”VersionNumber、目标“ItemID”、目标“Item”VersionNumber、LinkType、Link“ItemID”和链接“Item”VersionNumber。

如图 6 所示，项目也可以使用“项目引用”属性彼此关联，所述“项目引用”属性像“项目”中包含的其他属性那样维护。“项目引用”属性用户定义的属性，是在“ItemType”32 内作为“ComponentType”36 的一部分创建的。此属性具有由内容管理系统维护的特殊的数据模型。它包含指向目标“Item”32' 的引用 58。类似于链接，（图 5），引用 58 到目标“Item”32' 的绑定可以是版本特定的，也可以是独立于版本的。当指定了“项目引用”58 属性时，为被引用的项目定义了“引用的范围”和“删除语义”。范围指定被引用的项目可以属于哪一个（哪些）“ItemType”，而“删除语义”类似于由 RDBMS 维护的删除规则，以确保当被引用的“项目”将被删除时的引用完整性。也可以对“项目引用”属性 58 不进行范围限定，在这种情况下，被引用的“项目”可以属于任何“ItemType”。在优选实施例中，同类范围“项目引用”属性（即，当范围是单一的“ItemType”时）由 RDBMS 外键支持。不同类范围（即，当范围包括多个“ItemType”时）和没有范围限定的“项目引用”属性是由内容管理系统使用 RDBMS 控制进行维护的。

此物理数据模型支持的另一个特殊的数据模型是“资源引用”属性 59 的数据模型，也在图 6 中进行了说明。如图 6 所示，资源引

用 59 包含对由内容管理系统的资源管理器进行管理的资源 61 的引用。资源管理器是对特定类型的资源进行管理的特殊用途的服务器。这样的例子有：对大量的二进制对象进行管理的对象服务器，管理并提供流式对象（例如，音频和视频内容）的流式服务器。可扩展的内容管理系统能够容纳新的资源管理器，其中每一个资源管理器都是为管理特定类型的资源而专门设计的，具有其特定表示和方法，不一定能够由这里所描述的物理数据模型而模型化。由内容管理系统进行管理的非结构化数据主要由一个或多个这些资源管理器在内部进行管理。资源引用的完整性是由内容管理系统维护的。

这里所描述的物理数据模型能够支持各种各样的高级别的数据模型，涵盖不同的应用领域，并支持多个行业标准（例如，Dublin Core、XML）。同样重要的是，此物理数据模型还提供足够的灵活性以允许进行数据库优化，以便有效地支持高级别的数据模型。在一个示例中，高级别的数据模型使用一个“ItemType”的 Items 来表示独立的、可共享的图像，其中每一个图像都具有其自己的属性、版本和 ACL。实际图像是作为“资源”存储的。然后使用另一种“ItemType”的 Items 来表示文档，并使用链接来将每一个文档绑定到它包含的图像。在另一个示例中，为支持高性能的生产应用，使用了单一的“Item”来表示文档连同包含在文档中的图像。（实际图像也作为“资源”来存储。）在此情况下，性能是关键，但对于图像，没有共享要求。

在“ItemType”上可以定义“ItemView(s)”。“ItemView”是命名的模板，以将对“ItemType”的访问限制到某些“Component”表和属性。它基本上是“ItemType”中的对应的“Component”表的 RDBMS 视图的集合。

本发明的一个应用是具有分布式读卡器访问的多媒体分布式数据库管理系统，如在磁卡安全系统、信用卡或贷记卡或自动柜员机卡系统或航空常客系统中所使用的那样。图 7 显示了这种系统的一个版本。系统 10 是由可以被读卡器 70a 读取的磁卡或光卡 70b 上的身份标记（例如，加密数据）驱动的。

在读取卡 70b 之后，获取生物特征读数。典型的生物特征数据包括指纹读取器 70c 上的指纹、数码相机 70d 上的声印和采样、面部图像和眼睛图像等等。生物特征、音频和图像数据通常通过单独的服务器 72a、73a、74a 和 75a 进行访问并由这些服务器进行处理，并存储在单独的数据库 72b、73b、74b 和 75b 中，例如，指纹数据、语音数据和面部和眼睛图像数据作为多媒体数据对象存储在单独的指纹数据、语音数据和面部和眼睛图像数据库中，并以这样的属性作为数据模型（图像、图像数据的数学表达式、音频数据，音频数据的数学表达式），以及仓库地址（例如，URL 或 IP 地址），其中，数据是项目，数据模型的地址和定义是属性，提供给筛选器、保安人员、出纳员、外勤登轮业务员或强制执行专业人员是资源。

所述方法和系统在物理上是以程序产品的方式实现的。程序产品可以驻留在一台计算机上、或多台计算机上、或分布式服务器上、或磁盘上、或磁带上。在安装之前或可以安装在一台或多台计算机上之前，程序本身可以经过加密和/或压缩，如分发介质或分布式服务器的情况。

本发明的进一步的应用是以向导的形式存在的方法和程序产品（即，结构化的对话框系列，询问问题并使用回答或选择来产生结果），用于以内容架构和元数据填充多媒体内容管理系统，包括高级别的内容模型和低级别的物理模型。这是通过如下方式来实现的：就内容项目向用户提供查询，并基于最终用户的响应，就内容项目提供一个或多个连续的查询，并基于这些响应，确定项目的子组件和属性。

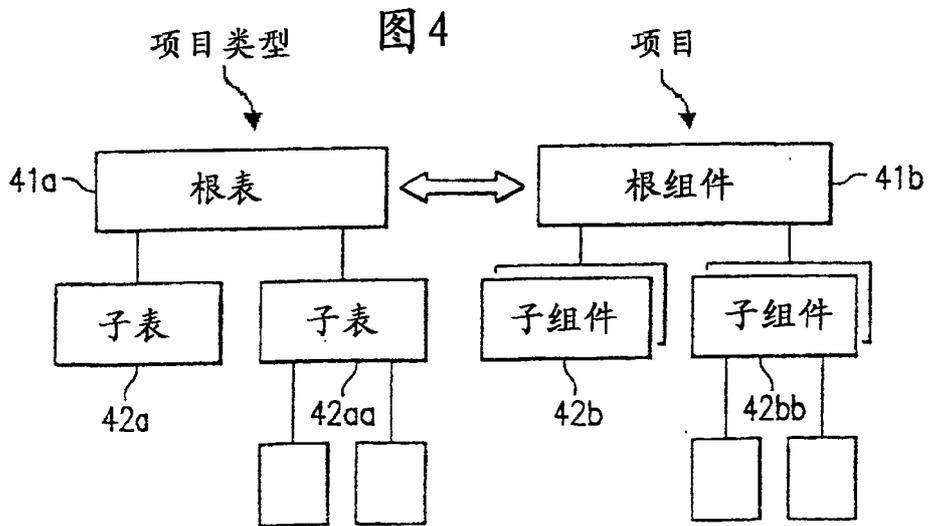
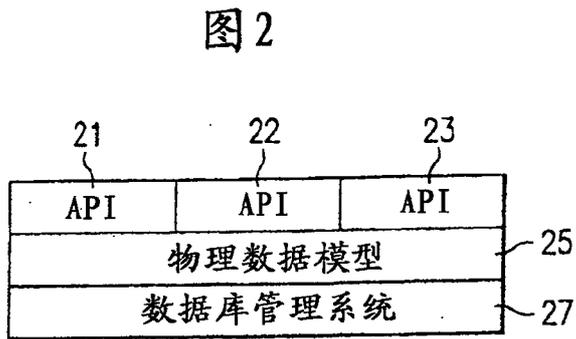
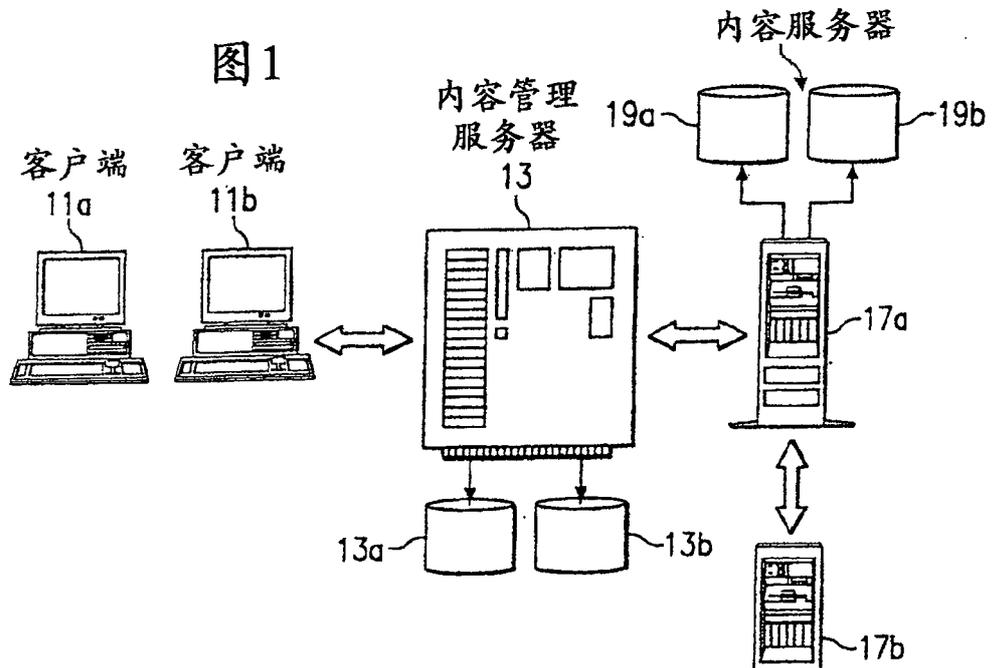


图3

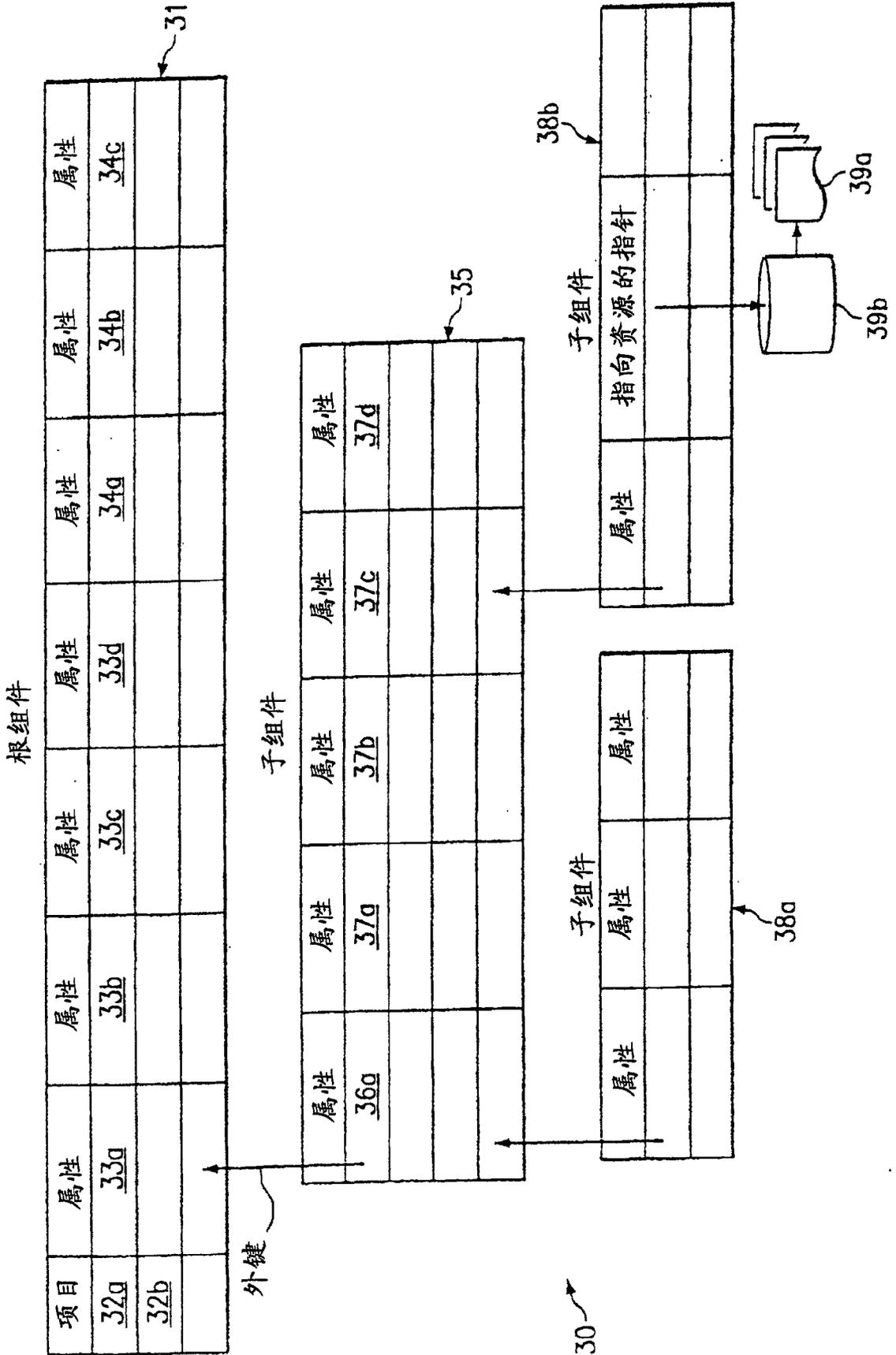


图5

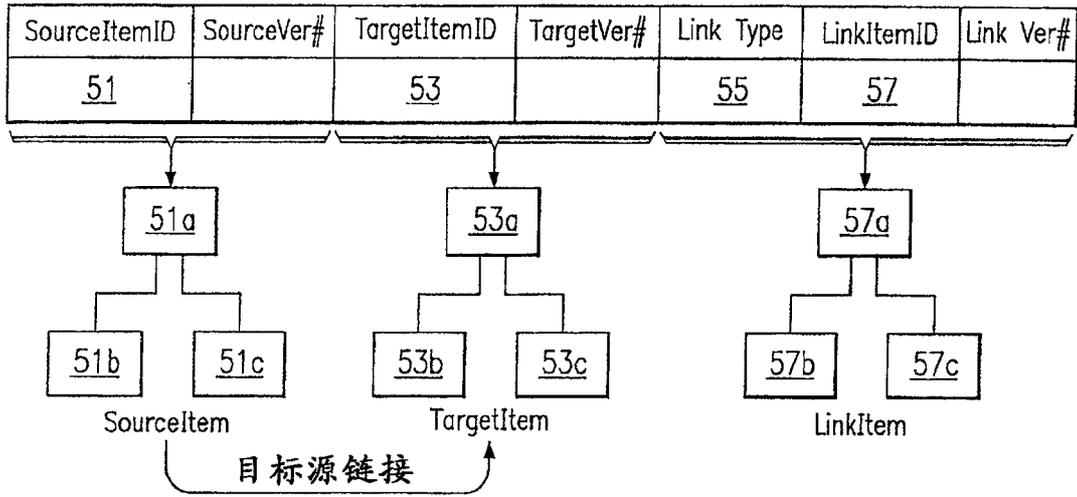


图6

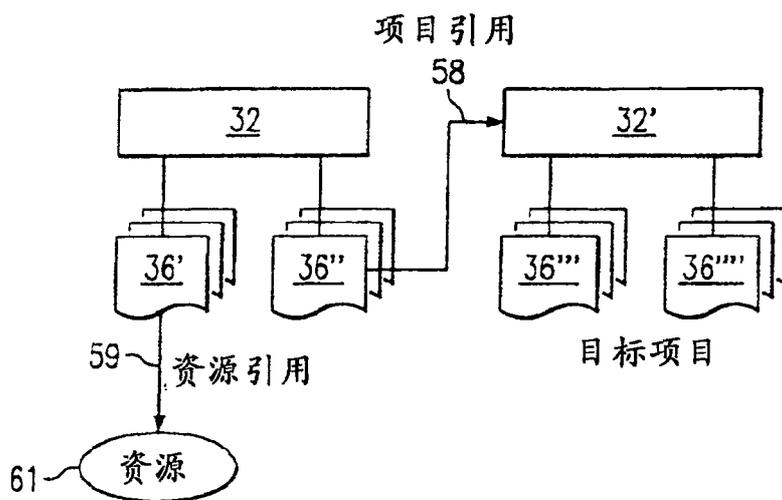


图7

