

[19] 中华人民共和国国家知识产权局

[51] Int. Cl.

G06F 17/30 (2006.01)



[12] 发明专利申请公开说明书

[21] 申请号 200510118862.5

[43] 公开日 2006 年 5 月 3 日

[11] 公开号 CN 1766886A

[22] 申请日 2005.10.25

[21] 申请号 200510118862.5

[30] 优先权

[32] 2004.10.25 [33] JP [31] 309439/04

[71] 申请人 惠普开发有限公司

地址 美国德克萨斯州

[72] 发明人 山本昭夫 清水裕之 鹫饲文敏

[74] 专利代理机构 中国专利代理(香港)有限公司

代理人 杨 凯 王 勇

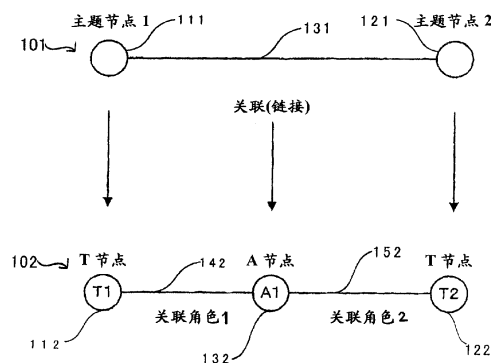
权利要求书 7 页 说明书 75 页 附图 52 页

[54] 发明名称

用于数据管理和/或转换的数据结构、数据库系统及方法

[57] 摘要

一种数据库系统，用于通过把一个或多个关联节点中的每个与一个或多个主题节点相关联来创建目录树形式的数据库，一个或多个主题节点中的每个具有属于其中的数据，以及定义关联属性来表示相关联的关联节点与主题节点之间的关联中每一个，所述数据库系统包括：目录树创建部件，用于创建关联节点中每个的关联节点条目以及关联属性中每个的关联属性条目，以及用于通过根据相应关联节点与相应关联属性之间的关联将条目相关来创建目录树；以及数据相关部件，用于根据主题节点与相应关联属性之间的关联把属于主题节点之一的数据与所创建关联属性条目相关。



1. 一种数据库系统，用于通过把一个或多个关关节点中的每个与一个或多个主题节点相关联来创建目录树形式的数据库，所述一个或多个主题节点中的每个具有属于其中的数据，以及定义关联属性来表示相关联的关关节点与主题节点之间的关联中的每个，所述数据库系统包括：

目录树创建部件，用于创建所述关关节点中的每个的关关节点条目以及所述关联属性中的每个的关联属性条目，以及用于通过根据相应关关节点与相应关联属性之间的关联将所述条目相关来创建目录树；以及

数据相关部件，用于根据主题节点与相应关联属性之间的关联把属于主题节点之一的数据与所创建的关联属性条目相关。

2. 如权利要求 1 所述的数据库系统，其特征在于，所述关联属性表示在相关联的关关节点与主题节点之间定义的角色。

3. 如权利要求 1 所述的数据库系统，其特征在于，所述目录树包括一个或多个子树，为所述目录树和所述子树定义所述目录树的顶部条目，所述子树中每个的至少一个顶部条目引用所述目录树和/或其它子树的条目中的至少一个，以及

所述关关节点和主题节点在所述目录树和所述子树中以分层方式相关。

4. 如权利要求 3 所述的数据库系统，其特征在于，还包括顶部条目相关部件，用于把所述子树中每个的至少一个顶部条目与由所述至少一个顶部条目引用的目录树和/或其它子树中的至少一个条目相关。

5. 如权利要求 3 所述的数据库系统，其特征在于，还包括一个或多个第一数据库服务器，用于存储属于目录树和子树中的一个中

包含的主题节点中至少一个的数据、属于与所述至少一个主题节点相关的关联节点的数据、以及根据目录树和子树中的所述一个在相关联的主题节点和关联节点之间定义的角色。

5 6. 如权利要求 5 所述的数据库系统，其特征在于，还包括数据传输部件，用于把一个所述第一数据库服务器中存储的数据传输并存储在另一个所述第一数据库服务器中。

7. 如权利要求 6 所述的数据库系统，其特征在于，所述数据传输部件根据从其中传输数据的第一数据库服务器的操作状态来确定要传输的数据。

10 8. 如权利要求 6 所述的数据库系统，其特征在于，所述数据传输部件根据对目录树或子树的访问状态来确定要传输的数据。

9. 如权利要求 5 所述的数据库系统，其特征在于还包括，
第二数据库服务器，用于存储第一数据库服务器中的一个或多个中存储的数据的复制品，以及

15 顶部条目相关部件，用于把子树中每个的至少一个顶部条目与由所述至少一个顶部条目引用的目录树和/或其它子树中的至少一个条目相关，以及还用于把第一数据库服务器的顶部条目与第二数据库服务器相关。

20 10. 如权利要求 9 所述的数据库系统，其特征在于，目录树和子树中的每个还包括一个或多个另外的子树，以及所述数据库系统还包括用于从目录树和子树中的每个中创建至少一个所述另外的子树的子树创建部件。

25 11. 如权利要求 9 所述的数据库系统，其特征在于，为目录树和子树中的每个定义用于对关联节点和主题节点分类的分类信息，以及所述数据库系统还包括用于把目录树和子树中的每个与为所述目录树或子树定义的分类信息相关的分类信息相关部件。

12. 如权利要求 11 所述的数据库系统，其特征在于，还包括检索装置，用于检索属于主题节点中至少一个的所存储数据。

13. 如权利要求 12 所述的数据库系统, 其特征在于, 所述检索装置包括

搜索条件接受部件, 用于接受搜索条件,

5 目录树搜索部件, 用于搜索与对应于所接受的搜索条件的分类信息相关的目录树或子树, 以及

数据搜索部件, 用于搜索属于由所述目录树搜索部件找到的目录树或子树中包含的主题节点中至少一个的所存储数据。

14. 一种数据管理方法, 用于通过把一个或多个关关节点中的每个与一个或多个主题节点相关联来创建目录树形式的数据库, 所述
10 一个或多个主题节点中的每个具有属于其中的数据, 以及定义关联属性来表示所述相关联的关关节点与主题节点之间的关联中的每个, 所述方法包括:

创建关关节点中的每个的关关节点条目以及关联属性中的每个的关联属性条目;

15 通过根据相应关关节点和相应关联属性之间的关联将所述条目相关来创建目录树; 以及

根据主题节点与相应关联属性之间的关联把属于主题节点之一的数据与所创建的关联属性条目相关。

15. 如权利要求 14 所述的数据管理方法, 其特征在于:

20 所述目录树包括一个或多个子树,

为目录树和子树定义目录树的顶部条目,

子树中的每个的至少一个顶部条目引用目录树和/或其它子树的条目中的至少一个, 以及

25 目录树和子树中的每个还包括一个或多个另外的子树; 以及

所述方法还包括从目录树和子树中的每个中创建至少一个所述另外的子树。

16. 如权利要求 14 所述的数据管理方法, 其特征在于:

所述目录树包括一个或多个子树,

为目录树和子树定义目录树的顶部条目, 以及
子树中的每个的至少一个顶部条目引用目录树和/或其它子树的
条目中的至少一个; 以及

所述方法还包括:

- 5 存储属于目录树和子树中的一个中包含的主题节点中的至少一个的数据、属于与所述至少一个主题节点相关联的关联节点的数据、以及根据目录树和子树中的所述一个在相关联的主题节点和关联节点之间定义的角色; 以及

搜索属于所述主题节点的所存储数据。

- 10 17. 如权利要求 16 所述的数据管理方法, 其特征在于还包括:
接受搜索条件;
搜索与对应于所接受的搜索条件的分类信息相关的目录树或子树; 以及

- 15 搜索属于作为搜索结果得到的目录树或子树中包含的主题节点的所存储数据。

18. 如权利要求 14 所述的数据管理方法, 其特征在于还包括:

- 存储属于目录树和子树中的一个中包含的主题节点中的至少一个的数据、属于与所述至少一个主题节点相关联的关联节点的数据、以及根据目录树和子树中的所述一个在多个第一数据库服务器中在
20 相关联的主题节点和关联节点之间定义的角色; 以及

将一个所述第一数据库服务器中存储的数据传输并存储在另一个所述第一数据库服务器中。

19. 一种其中存储程序的计算机可读介质, 所述程序用于包括计算机的数据库系统中, 用于通过把一个或多个关联节点中的每个与
25 一个或多个主题节点相关联来创建目录树形式的数据库, 所述一个或多个主题节点中的每个具有属于其中的数据, 以及定义关联属性来表示相关联的关联节点与主题节点之间的关联中的每个,

所述程序在所述数据库系统中运行时, 使所述计算机执行以下

步骤:

创建关联节点中的每个的关联节点条目以及关联属性中的每个的关联属性条目;

5 通过根据相应关联节点和相应关联属性之间的关联将所述条目相关来创建目录树; 以及

根据主题节点与相应关联属性之间的关联把属于主题节点之一的数据与所创建的关联属性条目相关。

20. 如权利要求 19 所述的计算机可读介质, 其特征在于:

10 所述目录树包括一个或多个子树,
为目录树和子树定义目录树的顶部条目,
子树中的每个的至少一个顶部条目引用目录树和/或其它子树的条目中的至少一个, 以及

目录树和子树中的每个还包括一个或多个另外的子树; 以及
15 所述程序在所述数据库系统中运行时还使所述计算机执行从目录树和子树中的每一个中创建至少一个所述另外的子树的步骤。

21. 如权利要求 19 所述的计算机可读介质, 其特征在于:

所述目录树包括一个或多个子树,
20 为目录树和子树定义目录树的顶部条目, 以及
子树中的每个的至少一个顶部条目引用目录树和/或其它子树的条目中的至少一个; 以及

所述程序在所述数据库系统中运行时, 还使所述计算机执行以下步骤:

25 存储属于目录树和子树中的一个中包含的主题节点中的至少一个的数据、属于与所述至少一个主题节点相关联的关联节点的数据、以及根据目录树和子树中的所述一个在相关联的主题节点和关联节点之间定义的角色; 以及

搜索属于所述主题节点的所存储数据。

22. 如权利要求 21 所述的计算机可读介质, 其特征在于, 所述

程序在所述数据库系统中运行时，还使所述计算机执行以下步骤：

接受搜索条件；

搜索与对应于所接受的搜索条件的分类信息相关的目录树或子树；以及

5 搜索属于作为搜索结果得到的目录树或子树中包含的主题节点的所存储数据。

23. 如权利要求 19 所述的计算机可读介质，其特征在于，所述程序在所述数据库系统中运行时，还使所述计算机执行以下步骤：

10 存储属于目录树和子树中的一个中包含的主题节点中的至少一个的数据、属于与所述至少一个主题节点相关联的关联节点的数据、以及根据目录树和子树中的所述一个在多个第一数据库服务器中在相关联的主题节点和关联节点之间定义的角色；以及

将一个所述第一数据库服务器中存储的数据传送并存储在另一个所述第一数据库服务器中。

15 24. 一种计算机可读存储器，其中存储了一种数据结构，所述数据结构包括：

主题节点字段，用于在其中存储属于多个对象的对象数据；

关联节点字段，用于在其中存储描述所述对象之间的关联的关联数据；以及

20 关联属性字段，用于在其中存储表示所述对象对于所述相关联的关联充当的角色的属性的角色数据。

25 25. 一种用于把具有由多个链接所连接的多个节点的第一数据结构转换为具有多个相关联的主题节点和关联节点的第二数据结构的

方法，所述方法包括：

把所述第一数据结构的节点中的每个转换为所述第二数据结构的主题节点之一；

把所述第一数据结构的链接中的每个转换为所述第二数据结构的关联节点之一；

在所述第二数据结构中，把关联节点中的每个与对应于在所述第一数据结构中由对应于所述关联节点的链接所连接的节点的主题节点中的两个相关联；以及

- 5 在所述第二数据结构中，把关联属性分配给相关联的关联节点与主题节点之间的每个关联，以表示所述节点对于所述第一数据结构中的相关链接所充当的角色的属性。

用于数据管理和/或转换的数据结构、数据库系统及方法

5 相关申请

本申请基于 2004 年 10 月 25 日提交的日本申请号 2004-309439 并要求其优先权，其公开通过引用完整地结合于本文中。

技术领域

10 本发明的实施例涉及用于数据管理和/或转换的数据结构、数据库系统、方法及存储程序的计算机可读介质。

背景技术

15 关系数据库一直用于存储相关数据以及用于搜索预期的存储数据。

实例包括以下参考文献：“相关数据模型白皮书”(Lazy Software, 2000 年 9 月), JP 2001-209647 A, 以及 WO 00/29980, 通过引用全部完整地结合到本文中。

20 但是,在上述参考文献中公开的方法的一部分中,不易改变已经完成的关系数据库的结构(模式)。

在上述参考文献中公开的其它方法中,数据的描述性内容变得复杂,以及数据不一定总是在存储之前被唯一地分析,因此原始数据可能不会在必要时从存储数据中准确再生。

25 此外,在上述参考文献中公开的方法中,当数据库分布在多个服务器当中时,不易在所述服务器当中分配负荷。

发明内容

根据本发明的一个实施例,提供一种数据库系统,用于通过把一

个或多个关联节点中的每个与一个或多个主题节点相关，来创建目录树形式的数据库，一个或多个主题节点中的每个具有属于其中的数据，以及关联属性被定义为表示相关的关联节点与主题节点之间的关联中的每一个。数据库系统包括：目录树创建部件，用于创建关联节点中每个的关联节点条目以及关联属性中每个的关联属性条目，以及
5 用于通过根据相应关联节点与相应关联属性之间的关联将这些条目相关来创建目录树；以及数据相关部件，用于根据主题节点与相应关联属性之间的关联把属于主题节点之一的数据与所创建的关联属性条目相关。

10 根据本发明的一个实施例，提供一种数据管理方法，用于通过把一个或多个关联节点中的每个与一个或多个主题节点相关联，来创建目录树形式的数据库，一个或多个主题节点中的每个具有属于其中的数据，以及关联属性被定义为表示相关的关联节点与主题节点之间的关联中的每个。该方法包括：创建关联节点中的每个的关联节点条目
15 以及关联属性中的每个的关联属性条目；通过根据相应关联节点与相应关联属性之间的关联将这些条目相关来创建目录树；以及根据主题节点与相应关联属性之间的关联把属于主题节点之一的数据与所创建的关联属性条目相关。

20 根据本发明的一个实施例，提供在其中存储程序的计算机可读介质。该程序用于包括计算机的数据库系统中，用于通过把一个或多个关联节点中的每个与一个或多个主题节点相关联，来创建目录树形式的数据库，一个或多个主题节点中的每个具有属于其中的数据，以及
25 关联属性被定义为表示相关的关联节点与主题节点之间的关联中的每个。该程序在数据库系统中运行时使计算机执行以下步骤：创建关联节点中的每个的关联节点条目以及关联属性中的每个的关联属性
条目；通过根据相应关联节点与相应关联属性之间的关联将这些条目相关来创建目录树；以及根据主题节点与相应关联属性之间的关联把属于主题节点之一的数据与所创建的关联属性条目相关。

根据本发明的一个实施例,提供一种在其中存储数据结构的计算机可读存储器。该数据结构包括:主题节点字段,用于在其中存储属于多个对象的对象数据;关联节点字段,用于在其中存储描述所述对象之间的关联的关联数据;以及关联属性字段,用于在其中存储表示
5 这些对象对于相关的关联所充当的角色的属性的角色数据。

根据本发明的一个实施例,提供一种方法,用于把具有由多个链接所连接的多个节点的第一数据结构转换为具有多个相关主题和关联节点的第二个数据结构。该方法包括:把第一数据结构的节点中的每个转换成第二个数据结构的主题节点之一;把第一数据结构的链接中的
10 每个转换成第二个数据结构的关联节点之一;在第二个数据结构中,把关联节点中的每个与对应于在第一数据结构中由与所述关联节点对应的链接所连接的节点的主题节点中的两个相关联;以及在第二个数据结构中,把关联属性分配给相关的关联与主题节点之间的每个关联,以便表示这些节点对于第一数据结构中的相关链接充当的角色的属性。
15 性。

通过思考以下具体结合附图对具体实施例的详细说明,本发明的实施例的目的、特征和优点将会非常明显。

附图说明

20 在附图的各图中,以举例而不是限制的方式说明本发明的实施例,整个附图中,具有相同参考标号的部件表示相似的部件。

图1是示意图,示意说明根据本发明的一个实施例转换相关网络数据时的数据结构或模型。

25 图2(a)是示意图,示意说明要存储的一系列(n)相关数据,以及图2(b)是示意图,示意说明如何根据图1所示的数据模型来存储所述数据系列。

图3(a)-(b)是示意图,均示意说明根据本发明的一个实施例、表达相关节点之间的关联的方法的一个实例。

图 4 是示意图, 示意说明根据本发明的一个实施例、表达相关网络数据的实例。

图 5 是本发明的一个优选实施例的框图。

图 6 是示意图, 说明根据本发明的一个实施例的第一数据库系统
5 (DB 系统) 的配置。

图 7 是示意图, 说明图 6 所示的 DB 服务器和 PC 的硬件配置。

图 8 是示意图, 示意说明重新排列形式的图 3(a) 的数据结构。

图 9 是示意图, 示意说明一般化形式的图 8 的数据结构。

图 10 是示意图, 示意说明一般化形式的图 3(b) 和图 4 的数据关
10 联。

图 11 是用于存储图 9 所示的数据结构的数据的关联角色(AR)表的视图。

图 12 是用于存储图 9 所示的数据结构的数据的 T 节点标识符(ID)表的视图。

图 13 是用于存储图 9 所示的数据结构的数据的 A 节点标识符(ID)表的视图。
15

图 14 是示意图, 示意说明根据本发明的一个实施例、在例如图 6 和图 7 所示的 DB 服务器中的数据搜索方法。

图 15 是流程图, 说明在图 6 和图 7 所示的 DB 服务器中执行的
20 搜索过程。

图 16 是流程图, 说明基于图 15 所示的搜索过滤器的相关节点选择过程。

图 17 是流程图, 说明获取图 15 和图 16 所示的节点 ID 和节点名称的过程。

图 18 是框图, 示意说明根据本发明的一个实施例、例如在图 6 和图 7 所示的 DB 服务器中运行的 DB 程序 2 的结构。
25

图 19 是示意图, 示意说明对图 6 和图 7 所示的 DB 服务器(DB 程序; 图 18)的数据输入以及搜索条件。

图 20 是根据本发明的一个实施例、例如由 AR 条目创建单元(图 18)和 ARDB 管理单元所创建并存储在 ARDB 中的 AR 表的视图。

图 21 是根据本发明的一个实施例、例如由 ID 条目创建单元(图 18)和 IDDB 管理单元所创建并存储在 IDDB 中的 T 节点 ID 表的视图。

图 22 是根据本发明的一个实施例、例如由 ID 条目创建单元(图 18)和 IDDB 管理单元所创建的 A 节点 ID 表的视图。

图 23 是示意图,说明根据本发明的一个实施例的第二 DB 系统的配置。

图 24 是示意图,说明本发明的一个实施例的第三 DB 系统的配置。

图 25 是示意图,示意说明图 24 所示的 DB 系统中的信息的图形表示。

图 26 是示意图,示意说明图 25 所示的信息的目录信息树表示。

图 27 是示意图,示意说明根据本发明的一个实施例分割以目录结构表示的相关数据的方法。

图 28 是示意图,示意说明图 27 所示的 DB 服务器(A)的条目 dn_A2 与 DB 服务器(N)的条目 dn_N 之间的引用关系。

图 29 是根据本发明的一个实施例、用于管理在图 27 所示的 DB 服务器(A 至 N)中分割及存储的相关数据的目录树表的视图。

图 30 是根据本发明的一个实施例、用于对 DB 服务器(A 至 N)中存储的目录树(或子树)的顶部条目 dn_A 至 dn_N 以及在子树中存储的相关数据进行分类的数据表的视图。

图 31 是示意图,示意说明根据本发明的一个实施例、在图 24 所示的 DB 管理服务器中运行的第一 DB 管理程序的结构。

图 32 是根据本发明的一个实施例、通过使用图 31 所示的 DB 管理程序在用于输入数据的计算机(PC)的输入/输出装置(图 6 和图 7)上显示的 GUI 屏幕的视图。

图 33 是流程图,说明根据本发明的一个实施例、采用图 31 所示的 DB 管理程序的相关数据的登记过程。

图 34 是流程图,说明根据本发明的一个实施例、采用图 31 所示的 DB 管理程序修改数据表(图 30)的过程。

5 图 35 是流程图,说明根据本发明的一个实施例、采用图 31 所示的 DB 管理程序修改目录树表的过程。

图 36 是顺序图,示意说明根据本发明的一个实施例的图 31 所示的 DB 管理程序的整体运行。

10 图 37 是示意图,示意说明根据本发明的一个实施例在图 24 所示的 DB 服务器的每个中运行的第二 DB 程序。

图 38 是示意图,示意说明根据本发明的一个实施例在图 24 所示的检索装置中运行的搜索程序。

图 39 是流程图,表示采用图 38 所示的搜索程序的搜索过程。

15 图 40 是顺序图,说明根据本发明的一个实施例、图 31 所示的 DB 管理程序以及图 38 所示的搜索程序的整体过程。

图 41 是第一示意图,示意说明根据本发明的一个实施例在图 24 所示的 DB 系统的 DB 服务器中登记的平面目录结构的相关数据。

图 42 是第二示意图,示意说明图 41 所示的目录结构的相关数据。

20 图 43 是第三示意图,示意说明图 41 所示的目录结构的相关数据。

图 44 是第四示意图,示意说明根据本发明的一个实施例、仅表示特定区域的销售信息、在图 24 所示的 DB 系统的 DB 服务器中登记的目录结构的相关数据。

25 图 45 是第五示意图,示意说明根据本发明的一个实施例、表示多个区域的销售信息、在图 24 所示的 DB 系统的 DB 服务器中登记的目录结构的相关数据。

图 46 是包含图 44 和图 45 所示的销售信息(相关数据)的顶部条目

的目录树表的视图。

图 47 是根据本发明的一个实施例、在图 44 和图 45 所示的销售信息(相关数据)被分割为子树时包含其顶部条目、分类类型和分类值的数据表的视图。

5 图 48 是根据本发明的一个实施例、用于搜索图 44 和图 45 所示的销售信息的 GUI 屏幕的视图。

图 49 是示意图, 示意说明根据本发明的一个实施例、由图 24 所示的检索装置执行以便检测要从图 46 和图 47 所示的目录树表和数据表中检索的 DB 服务器的过程。

10 图 50 是创建从图 48 所示的搜索条件中得到的并用于仅搜索图 44 所示的 Kanto 地区的销售信息的 LDAP 操作的搜索请求消息的视图。

图 51(A)-(B)是示意图, 均示意说明根据本发明的一个实施例、分别在新子树的添加之前和之后在图 24 所示的 DB 系统的 DB 服务器中新添的子树。

15 图 52 是用于图 51(A)和(B)所示的新子树的添加的搜索条件管理表的视图。

图 53(A)-(B)是示意图, 均示意说明根据本发明的一个实施例、分别在交换之前和之后在图 24 所示的 DB 系统的 DB 服务器之间的子树交换/移动。

20 图 54(A)-(B)是示意图, 说明根据本发明的一个实施例、分别在更新之前和之后、参与图 53(A)和(B)所示的子树交换的 DB 服务器的目录树表(图 46)的更新。

图 55 是示意图, 示意说明根据本发明的一个实施例的第三 DB 程序, 它在新子树的添加及其移动/交换在图 24 所示的 DB 系统中执行时在 DB 服务器中运行。

25 图 56 是示意图, 示意说明根据本发明的一个实施例的第二 DB 管理程序, 它在新子树的添加及其移动/交换在图 24 所示的 DB 系统中执行时在 DB 管理服务器中运行。

图 57 是流程图，说明图 56 所示的 DB 管理程序的过程。

图 58 是示意图，示意说明根据本发明的一个实施例、在多个 DB 服务器上的相同层上的条目。

5 图 59 是示意图，示意说明根据本发明的一个实施例的第二搜索程序，它在新子树的添加及其移动/交换在图 24 所示的 DB 系统中执行时在检索装置上运行。

具体实施方式

10 在详细说明本发明的实施例之前，大家要理解，本发明不限于它在以下描述中阐述或者在附图中所示的构造的详细情况以及组件配置的应用。本发明能够采用其它实施例并且能够以各种方式来实施或执行。另外还要理解，本文所使用的措辞和术语是为了便于描述而不应当被视作限制。用于标识方法或过程的步骤的字母只是用于标识而不是意味着表示这些步骤应当以特定顺序来执行。

15 包含各种成分的数据经过分析，并存储已分析数据。

希望在必要时，原始数据从存储数据中准确再生。

一般来说，在集合 $A \times B$ 的笛卡儿积的部分集合 $R \subseteq A \times B$ 中，对于有序对 $(a,b) \in R$ 的 aRb 表示法表示“a 与 b 具有关系 R”。作为需要被存储以及将来再生的原始数据的一个简单实例，现在考虑“剧作家莎士比亚写作戏剧哈姆雷特”。

20 原始数据处于二元关系。即，原始数据“剧作家莎士比亚写作戏剧哈姆雷特”可表达为“aRb”格式，其中，a 为“剧作家莎士比亚”，R 为“作者-作品”，以及 b 为“戏剧哈姆雷特”。

25 这样，原始数据作为“a”、“R”和“b”被存储在数据库中，并且可在必要时再生。

但是，当数据成分的数量增加时，即原始数据不是二元关系而是 n 元关系时，一系列这类相关数据以超图结构来表达，并且数据的处理不是简单的。

因此，采用一种把任何 n 元关系分解为多个二元关系、把数据分析为二元关系的组合以及在数据库中存储已分析数据的方法。

作为第一实例，现在将考虑“剧作家莎士比亚在大约 1600 年在英国写作戏剧哈姆雷特”。

- 5 这个实例表示一种四元关系，它被分解为六个二元关系的组合，如表 1 所示，因为 $4C2=6$ 。

表 1

a	R	b
剧作家莎士比亚	作者-作品	戏剧哈姆雷特
剧作家莎士比亚	作者-创作时间	约 1600 年
剧作家莎士比亚	作者-创作国家	英国
戏剧哈姆雷特	作品-创作时间	约 1600 年
戏剧哈姆雷特	作品-创作国家	英国
约 1600 年	创作时间-创作国家	英国

换言之，对于任何给定的 n ， $nC2$ 二元关系的组合是表达 n 元关系所必需的。

- 10 随后，将作为第二实例来考虑“剧作家莎士比亚在大约 1600 年写作戏剧第十二夜”。这个实例表示一种三元关系，它被分解为 3 个二元关系的组合，即 $n=3$ ，且 $3C2=3$ 。结果如表 2 所示。

表 2

a	R	b
剧作家莎士比亚	作者-作品	戏剧第十二夜
剧作家莎士比亚	作者-创作时间	约 1600 年
戏剧第十二夜	作品-创作时间	约 1600 年

- 15 在这种情况下，在相同数据库中第一实例的信息以及第二实例的信息的存储导致一个问题：由“剧作家莎士比亚”-“作者-创作时间”-“约 1600 年”组成的相同二元关系被存储。

另一个问题是判断用于表达第一和第二实例的数据的二元关系的组合中哪一个应当用来再生原始信息的不可能性。换言之，数据、例如第一和第二实例没有被唯一分析和存储。

5 这些问题可通过对二元关系中的每个添加标识符得到解决。但存在一个缺点，因为数据结构和处理将变得更复杂。

在关系数据库中存储二元或三元或者任何 n 元数据是已知的。

关系数据库包括其中存储数据项的一个或多个字段。关系数据库的一个简单实例是表格，其中，字段或数据元素以列排列，以及数据项以行排列。本文所使用的“字段”或“数据元素”被理解为数据的逻辑定义，而“数据项”则被理解为字段中存储的数据的实际单元。

10 现在回到第一实例“剧作家莎士比亚在大约 1600 年在英国写作戏剧哈姆雷特”。

(1) “谁”、(2) “什么”、(3) “何时”、(4) “何地”、(5) “为何”以及(6) “如何”可指定为数据元素或字段。

15 与这些字段相对应，可存储(1) “剧作家莎士比亚”、(2) “戏剧哈姆雷特”、(3) “约 1600 年”、(4) “英国”、(5) “空”以及(6) “写作”的数据项。

但是，下列问题是采用关系数据库存储数据的方法中固有的。

20 (1) 以后当新数据进入时，不易添加数据元素。在由定型成分组成的数据的情况中，添加不成问题。但是，在包含各种成分的输入数据的情况下，每当带有一个或多个新数据成分的新数据进入时，需要改变数据库的模式以添加数据元素。模式改变一般是困难的，特别是在联机输入数据时。

25 (2) 由于不易在以后添加新的字段或数据元素，因此在初始数据库构建中，模式可采用最大数量的字段或数据元素来构建。但是，字段中的许多不可能包含任何数据项。这不可避免地导致存储器使用效率的降低。

为了解决这类问题，已经提出基于“Lazy Software” Inc.(英国)

开发的“相关数据模型”的数据存储方法，作为替代传统关系数据库模型的模型。

按照这个相关数据模型，根据对象之间的关联来处理信息，以及这种关联以“源-动词-目标”语法来表达。

5 这个模型解决了基于关系数据库的存储方法中固有的一部分问题。

但是，该相关数据模型具有下列问题。在处理复杂数据关系的情况下，表达数据之间的关系的模型的方法是复杂且不直观的。如上所述，以二元树的形式表达 n 元关系时，数据没有被唯一分析和存储。
10 另外，当数据存储 in 数据库中时，操作员能够随意地改变数据结构。这样，原始信息可能不会准确地再生。

如上所述，关系数据库在传统上已知为存储彼此相关的多个数据的方法。

根据这个方法，字段或数据元素经过预置，以及数据项存储在这些
15 这些字段中。

该方法有利之处在于，数据关系可易于理解和/或表达。但是，不易对已经构建的数据库添加新字段或数据元素、即修改已经构建的数据库的结构(模式)。

如果将来添加新字段或数据项，则将对于存储数据产生“空”条
20 目，导致与存储器使用效率的降低有关的问题。

另一方面，在 Lazy Software Inc.提出的“相关数据模型”的情况下，与添加新数据项的困难以及存储器使用效率降低有关的问题得到解决。但是，数据描述内容变得复杂，以及数据结构不直观，从而导致所分析/存储数据不唯一的另一个问题。

25 图 1 说明根据本发明的一个实施例、由主题节点 111、121 以及表示节点 111、121 与数据模型 102 之间关联的链接 131 组成的相关网络数据 101。关联(链接)131 在数据模型 102 中重新定义为关联节点 132(以下称作“A 节点”)。主题节点 111、121 对应于数据模型 102

中的主题节点 112、122(以下称作“T 节点”)。数据模型 102 还包括分别把 T 节点 112、122 连接到 A 节点 132 并具有反映主题节点 111、121 对于相关网络数据 101 中的关联或链接 131 充当的角色(以下称作“关联角色”)的属性的链接 142、152。

5 这样, 相关网络数据 101 转换为数据模型 102。在本发明的一个实施例中, 数据模型 102 被重新构建。在一个简单实例中, 数据模型 102 以包括三个字段、即一个“A 节点”字段、一个“T 节点”字段和一个“关联角色”字段的关联角色表的形式来表示, 以及从相关网络数据 101 中提取的信息被作为数据项输入如表 3 所示的这种关联角色表(以下称作“AR 表”)的行(记录)中。在一个实施例中, 定义为关系数据库的一部分的 AR 表采用关系数据库管理系统来管理。

表 3

A 节点	T 节点	关联角色
A1	T1	关联角色 1
A1	T2	关联角色 2

15 因此, 与某些数据(主题节点)有关的新属性信息被定义为另一个相关数据, 并通过采用相应 A 和 T 节点以及节点之间的链接的组合(即根据本发明的一个实施例的数据模型中的基本成分)表达为与 AR 表的一行相关的数据, 从而可添加新属性信息而不改变现有表格结构(数据库模式)。

此外, 把标识符分配给 A 和 T 节点, 从而唯一标识这些节点。对于标识符, 标识符表(以下称作“ID 表”)被定义, 并且包含表示节点属性类型的节点类型以及表示属性值、即节点的具体内容的节点名称(表 5)。

ID 表由关系数据库管理系统来管理, 如在 AR 表的情况中那样。

20 在本发明的一个实施例中, 作为描述由某个 A 节点表示的关联的具体含义的数据来新添加 T 节点, 这两个节点、即预先存在的 A 节点和新的 T 节点通过预定义为“具体化”的关联角色彼此相关联。

通过进一步定义/描述新的 T 节点与类似地对另一个 A 节点添加的另一个新的 T 节点之间的关联, 能够表达两个原始 A 节点所表示的关联之间的关系。

5 新的 T 节点以及特别引入以描述 A 节点的含义的关联角色“具体化”可通过 AR 表来存储/管理。

通过采用 ID 表和 AR 表来管理节点, 实现能够不仅表达 A 与 T 节点之间的关联、而且表达 A 节点之间的关联的数据表达方法。

在先有方法中, nC2 二元关系是表达一段 n 元数据所必需的。但是, 根据本发明的一个实施例, 表达相同的 n 元数据只需要 n 个关系。

10 换言之, 根据本发明的一个实施例, 当由具有一个公共关联的 n 个数据成分组成的数据集(图 2, (a))被存储时, 数据集公共的一个新节点(A 节点)被添加, 然后对于每个数据成分、即 T 节点定义关联角色(图 2, (b))。

所得 AR 表由表 4 给出。

15

表 4

A 节点	T 节点	关联角色
A1	T1	关联角色 1
A1	T2	关联角色 2
A1	T3	关联角色 3
A1	T4	关联角色 4
:	:	:
A1	T(n-1)	关联角色(n-1)
A1	Tn	关联角色 n

图 2(b)和表 4 中采用的 A1 是分配给相应 A 节点的标识符, 表明数据成分(T 节点具有分配到其中的其它标识符“T1”至“Tn”)具有某种公共关联。

20 当数据成分具有多个不同的公共关联时, 具有多个不同标识符的多个 A 节点被添加。

此外，对于已经分配了标识符的 A 和 T 节点，创建具有作为数据属性的“节点类型”字段和“节点名称”字段的 ID 表，如表 5 所示。

表 5

节点 ID	节点类型	节点名称
A1	A 节点类型 1	A 节点名称 1
T1	T 节点类型 1	T 节点名称 1
T2	T 节点类型 2	T 节点名称 2
:	:	:
Tn	T 节点类型 n	T 节点名称 n

5 图 3(a)-3(b)说明一个实例，其中 T 节点 T1 和 T2 通过 A 节点 A1 彼此相关，以及 T 节点 T1 和 T3 通过 A 节点 A2 彼此相关(图 3,(a))，新增 T 节点(标识符 T11)以描述由 A 节点 A1 所示的关联的具体含义，并根据预定义为“具体化”的关联角色与 A 节点 A1 相关。

10 类似地，A 节点 A2 根据关联角色“具体化”与新的 T 节点 T12 相关，以及两个 T 节点 T11 与 T12 之间的关联通过采用 A 节点 A11 来定义(图 3，(b))。

因此，两个原始 A 节点 A1 和 A2 之间的关系可通过采用与表 6 所示类似的 AR 表来表达。

表 6

A 节点	T 节点	关联角色
A1	T11	具体化
A2	T12	具体化
A11	T11	关联角色 11
A11	T12	关联角色 12

15 数据存储方法/数据结构

下面将描述根据本发明的一个实施例的数据存储方法和数据结构。

作为一个具体实例，下面将考虑“剧作家莎士比亚在大约 1600 年在英国写作戏剧哈姆雷特”作为第一数据。

- 第一数据表示具有以下数据成分的四元关系：(1)“剧作家莎士比亚”、(2)“戏剧哈姆雷特”、(3)“约 1600 年”以及(4)“英国”。
- 5 当第一数据采用先有方法分析为二元关系时，这将需要六个二元关系的组合，如表 7 所示，因为 $4C2=6$ 。

表 7

主题节点 1	链接(关联)	主题节点 2
剧作家莎士比亚	作者-作品	戏剧哈姆雷特
剧作家莎士比亚	作者-创作时间	约 1600 年
剧作家莎士比亚	作者-创作国家	英国
戏剧哈姆雷特	作品-创作时间	约 1600 年
戏剧哈姆雷特	作品-创作国家	英国
约 1600 年	创作时间-创作国家	英国

已分析数据根据本发明的一个实施例来转换。表 7 的第一记录(行)按照以下所述来转换。

- 10 主题节点 1 中的数据成分“剧作家莎士比亚”在这个信息中表明“剧作家”。因此，数据成分被分析为“剧作家”和“莎士比亚”，其中“莎士比亚”被设置为 T 节点，以及“作者”被设置为关联角色。如以下所述，“剧作家”被设置为 T 节点。

- 15 对于表明关联的链接“作者-作品”，对于“哈姆雷特的出处”添加 A 节点，从而表明信息系列属于同一组。

由于“哈姆雷特的出处”用来表明属于同一组的信息系列，因此，只要此信息可与另一组的信息区分开，则允许其它表达。

这样，表 7 的第一记录(行)如表 8 所示进行转换。

表 8

A 节点	T 节点	关联角色
哈姆雷特的出处	莎士比亚	作者

类似地，主题节点 2 中的“戏剧哈姆雷特”按照表 9 所示进行转换，因为它在这个信息中是“作品”。另外，“戏剧”被设置为 T 节点。

表 9

A 节点	T 节点	关联角色
哈姆雷特的出处	哈姆雷特	作品

5 随后，表 7 的第二记录(行)的类似转换示于表 10。

表 10

A 节点	T 节点	关联角色
哈姆雷特的出处	莎士比亚	作者
哈姆雷特的出处	约 1600 年	创作时间

在这里，数据项“哈姆雷特的出处”、“莎士比亚”和“作者”可省略，因为它们是冗余的。此后，表 7 的全部记录(行)类似地经过转换，冗余数据被省略，结果如表 11 所示。

10

表 11

A 节点	T 节点	关联角色
哈姆雷特的出处	莎士比亚	作者
哈姆雷特的出处	哈姆雷特	作品
哈姆雷特的出处	约 1600 年	创作时间
哈姆雷特的出处	英国	创作国家

因此，当由具有一个公共关联的四个数据成分组成的数据集采用先有方法表达成二进制关系时，需要六个记录，因为 $4C2=6$ 。但是，大家会理解，根据本发明的一个实施例，仅需要四个记录。

15

在一个实施例中，当具有四个数据成分的数据集被分析并存储在数据库中时，添加数据集公共的一个新节点(A 节点)，然后对每个数据成分定义关联角色，以及定义具有“A 节点”、“T 节点”和“关联角色”字段的数据库。这样，根据本发明的实施例的数据结构可重新构建。

在这里，标识符“A1”分配给A节点“哈姆雷特的出处”。大家会理解，具有公共标识符“A1”的数据属于一组。另外，标识符“T11”至“T14”分别分配给四个T节点“莎士比亚”、“哈姆雷特”、“约1600年”以及“英国”。

5 这样，对于第一数据创建了AR表，如表12所示。

表 12

A 节点	T 节点	关联角色
A1	T11	作者
A1	T12	作品
A1	T13	创作时间
A1	T14	创作国家

表明“哈姆雷特的出处”的A节点(标识符A1)被设置为“出处相关信息”，以及已经分配了标识符T11至T14的T节点的节点类型被设置为“剧作家”、“戏剧”、“时间”和“国家”。因此，创建ID表，如表13所示。

10

表 13

节点 ID	节点类型	节点名称
A1	出处相关信息	(空)
T11	剧作家	莎士比亚
T12	戏剧	哈姆雷特
T13	时间	约 1600 年
T14	国家	英国

作为第二数据，将在另一个实例中考虑陈述“原著戏剧哈姆雷特的日译本由OO出版社于2003年2月出版”。

根据本发明的一个实施例，第二数据按照以下方式进行分析。在这里，所有数据成分公共的A节点是“哈姆雷特的日译本”，以及标识符“A2”分配到其中。结果如表14所示。

15

表 14

A 节点	T 节点	关联角色
A2	哈姆雷特	原著
A2	哈姆雷特	译本
A2	2003 年 2 月	出版日期
A2	OO 出版社	出版

标识符“T12”已经分配给作为原著的哈姆雷特。这样，当标识符“T22”被分配给译本哈姆雷特，以及标识符“T23”和“T24”被分配给出版日期和出版商时，第二数据的 AR 表被表示于表 15 中。

表 15

A 节点	T 节点	关联角色
A2	T12	原著
A2	T22	译本
A2	T23	出版日期
A2	T24	出版

5 另外，还创建 ID 表，如表 16 所示。

表 16

节点 ID	节点类型	节点名称
A2	出处相关信息	(空)
T22	戏剧	哈姆雷特
T23	日期	2003 年 2 月
T24	出版商	OO 出版社

第一和第二数据以及其它这类数据存储在相同数据库中。这样，最终获得与表 17 和表 18 类似的 AR 表和 ID 表。

表 17

A 节点	T 节点	关联角色
A1	T11	作者
A1	T12	作品
A1	T13	创作时间

A1	T14	创作国家
A2	T12	原著
A2	T22	译本
A2	T23	出版日期
A2	T24	出版
:	:	:

表 18

节点 ID	节点类型	节点名称
A1	出处相关信息	(空)
T11	剧作家	莎士比亚
T12	戏剧	哈姆雷特
T13	时间	约 1600 年
T14	国家	英国
A2	出处相关信息	(空)
T22	戏剧	哈姆雷特
T23	日期	2003 年 2 月
T24	出版商	OO 出版社
:	:	:

此外，如图 4 所示，提供附加的 T 节点以便描述具有标识符 A1 和 A2 的 A 节点所表明的关联的具体含义。标识符“T31”和“T32”被分配给附加/新的 T 节点，以及附加/新的 T 节点通过关联角色“具体化”与节点 A1 和 A2 关联。

这两个新的 T 节点的节点类型被设置为“出处信息”，以及节点名称分别被设置为“哈姆雷特的出处”和“哈姆雷特的日译本”。

新增表明节点 T31 与 T32 之间的关联的 A 节点，采用标识符 A3，以及节点类型被设置为“原著-译本信息”。

关联中节点 T31 和 T32 的角色为“原著信息”和“译本信息”。通过上述处理，添加了 AR 表和 ID 表，分别如表 19 和表 20 所示。在本发明的一个实施例中，表 19、表 20 分别附加到表 17、表 18。

表 19

A 节点	T 节点	关联角色
A1	T31	具体化
A2	T32	具体化

A3	T31	原著信息
A3	T32	译本信息

表 20

节点 ID	节点类型	节点名称
T31	出处信息	哈姆雷特的出处
T32	出处信息	哈姆雷特的日译本
A3	原著-译本信息	(空)

5 为了在相同关系数据库中直接存储第一和第二数据，必须新增例如与“哈姆雷特”对应的、作为翻译版本或“OO 出版社”的项(数据属性)作为新字段。根据先有方法，这种添加不容易，因为数据库的表结构必须相应改变。

但是，根据本发明的一个实施例，通过如上所述对现有 AR 表添加新的行、即新的记录而不是新的字段，通过把具有不同数据属性的数据作为不同组来存储，就能够区分第一数据和第二数据。

10 采用 A 节点、T 节点以及表示这些节点之间关联的链接来表达第一和第二数据，如图 4 所示。

本发明的一个实施例提供易于表示具有复杂结构的数据的方法以及采用关系数据库的存储/管理方法。

随后将描述对于本发明的一个实施例的数据库中的预期数据的示范搜索。

15 例如，用户希望知道剧作家莎士比亚创作的戏剧哈姆雷特的日译本的出版商。

图 5 是流程图，说明根据本发明的一个实施例执行的搜索过程。下面将说明图 5 的流程图。

20 在步骤 110: 用户输入搜索条件，例如由“剧作家”(或“作者”)、“莎士比亚”写作的“戏剧”(或“作品”)。

在步骤 120: 满足搜索条件的一组或多组数据被检索。

在步骤 130: 与所检索的一组或多组数据中的“戏剧”对应的数

据被显示。

在步骤 140: 用户从所显示数据中选择预期戏剧名称, 即“哈姆雷特”。

5 在步骤 150: 再次搜索数据库, 采用“哈姆雷特”和“译本”作为新的搜索条件。

在步骤 160: 满足新的搜索条件的另一组或多组数据被检索。

在步骤 170: 与所检索的一组或多组数据中的“出版商”和“出版日期”有关的数据被显示。

在步骤 180: 用户从所显示数据中选择预期出版商。

10 下面进行详细描述。

用户输入“莎士比亚”和“作者”作为搜索条件, 并搜索数据。

在这个特定实例中, 在数据库中存在 T 节点的节点名称“莎士比亚”和关联角色“作者”。通过参考数据库中的 AR 表(例如表 17)来检索其关联角色为“作者”的 T 节点的标识符(例如 T11)。随后, 通过参考 ID 表(例如表 18)中存储的节点名称属性来检索与其节点名称为“莎士比亚”的 T 节点(例如 T11)对应的一组或多组 A 节点标识符。

在包括所检索的一组或多组 A 节点标识符的搜索结果中, 从 AR 表(例如表 17)中选择另一个搜索条件、即其关联角色为“作品”的 T 节点标识符。

20 根据所选标识符从 ID 表(例如表 18)中显示与所选 T 节点、即与关联角色“作品”对应的戏剧有关的数据、即节点名称。

例如显示戏剧或节点名称“哈姆雷特”、“驯悍记”、“威尼斯商人”、“仲夏夜之梦”、“李尔王”等。用户选择预期戏剧或节点名称, 即“哈姆雷特”。

25 随后, 使用所选节点名称“哈姆雷特”的标识符“T12”作为关键字, 根据本发明的一个实施例的系统从 AR 表(例如表 17)中搜索包含标识符“T12”作为 T 节点 ID、同时具有关联角色“译本”的 A 节点标识符。

相应地，与满足搜索条件的 A 节点标识符相关的一组或多组数据被检索。

通过参考 ID 表(例如表 18)，从所检索的一组或多组数据中显示具有“出版”和“出版日期”作为关联角色的 T 节点的节点名称。

5 用户这时可从所显示数据中选择预期出版商，即实际上在“2003 年 2 月”新近出版了戏剧哈姆雷特的译本的“OO 出版社”。

作为一个搜索实例描述了图 5 的过程。毫无疑问，当搜索条件的数量增加时，在数据库中的搜索次数不限于如图 5 所示的两次，而是根据搜索条件可以是任意的次数。

10 在所述具体实例中，单一属性被定义为关联角色。但是，每个关联角色不限于只是一个单一属性。

换言之，关联角色可具有多个属性。在以上实例中，更详细的属性可通过把诸如“悲剧”、“喜剧”、“传奇”、“历史剧”等题材添加及指定到关联角色“戏剧”来定义。

15 数据存储方法/数据结构的特征

如上所述，根据本发明的一个实施例的数据存储方法和数据结构，通过采用广泛使用的关系数据库形式，可存储和管理数据，同时维护表示具有三元或更复杂关系、一般为 n 元相互关系的一系列数据的关系的超图结构。

20 按照根据本发明的一个实施例在关系数据库的表中直接映射相关网络数据的方法，能够解决无法有效地存储/管理具有三元或更复杂关系、一般为 n 元关系的数据的先有问题。

还能够解决以下先有问题：诸如把属性数据添加到数据库中存储的数据之类的修改必需改变关系数据库表的结构、导致灵活性损失以及要求更多工作量。

25 此外，能够通过利用通常分配给数据的附加标识符以及相同数据库模式的构架中的附加关联角色，为一系列关联提供具体含义。

第一数据库系统

下面将描述本发明的一个实施例的第一数据库。

图 6 说明实现本发明的一个实施例的第一数据库系统(DB 系统)1 的配置。

参照图 6, 本发明的一个实施例的第一 DB 系统 1 通过经由诸如
5 LAN、WAN 或因特网的网络 100 把数据库服务器(DB 服务器)12 连接到用于输入和搜索数据的计算机(PC)102 来配置。

在以下描述中, 术语可能部分或者略微不同于以上参照图 4、图 5 以及表 7 至表 20 描述数据存储方法和数据结构所使用的术语。但是, 描述的相应术语的含义实质上相同。

10 在出现矛盾时, 将以 DB 系统 1 的以下描述中使用的术语的含义为准。

在以下参考的附图中, 相似组件将由相似参考标号表示。

硬件配置

图 7 说明图 6 所示的 DB 服务器 12 和 PC 102 的硬件配置。

15 参照图 7, DB 服务器 12 和 PC 102 包括: 主单元 120, 其中包括 CPU 122、存储器 124 及其外围电路; 输入/输出装置 126, 其中包括显示单元和键盘; 以及记录装置 128, 例如 CD 或 HDD。此外, 当 DB 服务器 12 和 PC 102(用于进行通信的组件以下可通称为通信节点) 连接到网络 100 时, 可添加用于经由网络 100 与其它通信节点进行通信的通信装置 132。
20

换言之, DB 服务器 12 和 PC 102 包括作为配备了与其它通信节点进行通信的功能的计算机的组件。

数据结构

25 DB 服务器 12 构建为允许根据以上参照图 4、图 5 和表 7 至表 20 所述的本发明的一个实施例的数据存储方法及数据结构来存储数据以及搜索所存储的数据。

现在描述 DB 服务器 12 中的数据结构和数据搜索机制。

图 8 说明重新排列形式的图 3(a)的数据关联。

参照图 8, 在 DB 服务器 12 中, 主题节点(以下称作 T 节点)与一个或多个关联节点(以下称作 A 节点)相关, 以及在相关的 T 和 A 节点之间定义关联属性 R。

5 关联属性 R 可以是用于定义 T 与 A 节点之间的关联的任何属性。但是, 为了以下提供的具体清楚的描述, 将考虑一个具体实例, 其中关联属性 R 为关联角色 R(如本发明的一个实施例的数据存储方法和数据结构的以上描述中的详细说明)。

图 3(a)所示的数据关联可重新排列, 如图 8 所示。

10 在图 8 所示的 A 节点 A1 至 A_n(n 为 1 或大于 1 的整数, 但 n 在所有情况中不表示相同的数)中, A 节点 A1 以及与 A 节点 A1 相关联的 T 节点 T1-1 至 T1-3 和 T2-1 通过链接互连。

类似地, 与 A 节点 A2 相关联的 T 节点 T2-1、T2-2 和 T_n-1 与 A 节点 A2 通过链接互连。

15 对于 A 节点 A_n, 情况也是这样。与 A 节点 A_n 相关联的 T 节点 T_n-1 至 T_n-4 与 A 节点 A_n 通过链接互连。

换言之, 图 8 表明, T 节点 T2-1 具有与 A 节点 A1 和 A2 两者的关联, 以及 T 节点 T_n-1 具有与 A 节点 A2 和 A_n 两者的关联。

图 9 说明一般化形式的图 8 的数据结构。

20 图 8 中, 从 T 节点 T1-1 通过 A 节点 A1、T 节点 T2-1、A 节点 A2 和 T 节点 T_n-1 到 A 节点 A_n 的链接 851、852、853、854、855 沿着图 9 中从上到下方向延伸的路径 961 来表示。

在一个具体实例中, 图 9 还表明下列情况。

25 (1) T 节点 T1-1 至 T1-m1 和 T2-1 与 A 节点 A1 相关联, 以及在 T 节点 T1-1 至 T1-m1 和 T2-1 与 A 节点 A1 之间的关联(链接)中定义关联角色 R1-1 至 R1-m1 和 R1-0。

(2) T 节点 T2-1 至 T2-m2 以及从图 9 中省略的 T 节点与 A 节点 A2 相关联, 以及在 T 节点 T2-1 至 T2-m2 与 A 节点 A2 之间的关联中定义关联角色 R2-1 至 R2-m2。

(3)类似地,此后,从图9中省略的T节点与A节点相关联,以及在其之间定义关联角色R。

(4) T节点 T_{n-1} 至 T_{n-mn} 以及从图9中省略的T节点与A节点 A_n 相关联,以及在T节点 T_{n-1} 至 T_{n-mn} 与A节点 A_n 之间的关联中定义关联角色 R_{n-1} 至 R_{n-mn} (m_1 至 mn , n 为整数)。

换言之,在DB服务器12中,T节点中的每个与一个或多个A节点相关联,以及A节点中的每个与一个或多个T节点相关联,从而多个T节点可经由A节点彼此相关联,以及多个A节点可经由T节点彼此相关联。

在DB服务器12中,如图9所示相关联的A和T节点的多个组合可被存储。

图10是示意图,说明一般化形式的图3(b)和图4的数据关联。

参照图10,与A节点 A_1 相关联的T节点 T_{1-1} 至 T_{1-3} (以及 T_{3-1}) 经由链接连接到A节点 A_1 ,与A节点 A_2 相关联的T节点 T_{2-1} 至 T_{2-3} (以及 T_{3-2}) 经由链接连接到A节点 A_2 ,以及A节点 A_n 和T节点 T_{n-1} 至 T_{n-3} 和 T_{2-3} (以及 T_{3-n}) 经由链接连接在一起。

T节点 T_{2-3} 通过链接连接到A节点 A_2 和 A_n 这两者,这意味着T节点 T_{2-3} 与A节点 A_2 和 A_n 这两者相关联。

当如在这种情况下,通过A节点 A_1 、 A_2 和 A_n 相关联的一系列信息具有公共关联时,可定义新的关联节点 A_3 。

例如,当A节点 A_1 是与哈姆雷特的原著有关的信息,A节点 A_2 是与哈姆雷特的译本有关的信息,以及A节点 A_n 是与哈姆雷特的演出有关的信息时,A节点 A_1 、 A_2 和 A_n 所表示的相关数据具有作为有关哈姆雷特的数据的共同性。

因此,为了表明通过A节点 A_1 、 A_2 和 A_n 相关联的数据具有共同性,新的关联节点 A_3 被定义并存储在数据库中。

如图10中的虚线所示,为了具体描述A节点 A_1 和T节点 T_{1-1} 至 T_{1-3} 所表示的一系列信息,新的T节点 T_{3-1} 被定义并存储在数据

库中。

类似地,为了具体描述 A 节点 A2 和 An 的关联,新的 T 节点 T3-2 和 T3-n 被定义并存储在数据库中。

5 例如,数据“哈姆雷特的出处”被定义为 T 节点 T3-1 中的主题内容,数据“哈姆雷特的日译本”被定义为 T 节点 T3-2 中的主题内容,以及数据“哈姆雷特的演出”被定义为 T 节点 T3-n 中的主题内容。这些数据存储在数据库中。

另外,在新的 A 节点 A3 与 T 节点 T3-1 至 T3-n 的每个之间定义关联角色 R,并将其存储在数据库中。

10 例如,“原著信息”被定义为新的 A 节点 A3 与 T 节点 T3-1 之间的关联角色 R,“译本信息”被定义为新的 A 节点 A3 与 T 节点 T3-2 之间的关联角色 R,以及“演出信息”被定义为新的 A 节点 A3 与 T 节点 T3-n 之间的关联角色 R。这些数据存储在数据库中。

类似地,例如,在 A 节点 A1、A2 和 An 与 T 节点 T3-1、T3-2 和 T3-n 之间定义由系统预定义为“具体化”的关联角色 R。

图 11 说明用于存储图 9 所示的结构的数据的关联角色(AR)表。

图 12 说明用于存储图 9 所示的结构的数据的 T 节点标识符(ID)表。

图 13 说明用于存储图 9 所示的结构的数据的 A 节点标识符(ID)表。

在 DB 服务器 12 中,通过图 9 所示的结构相关的 A 和 T 节点的数据以及 T 节点的数据通过采用图 11 的 AR 表和图 12 的 ID 表来存储。

25 图 11 所示的 AR 表的条目(或记录或者行)中的每个表示一个给定的 A 节点、与 A 节点相关联的一个 T 节点以及在相关联的 A 与 T 节点之间定义的关联角色 R,并且包括 A 节点 ID、T 节点 ID 以及关联角色 R。

换言之,AR 表的条目中的每个包括图 9 所示的链接之一的一端

的 A 节点的标识符、链接的另一端的 T 节点的标识符以及定义链接的属性的关联角色。

对于图 9 所示的所有链接(T 节点 T1-1 与 A 节点 A1 之间以及 T 节点 Tn-mn 与 A 节点 An 之间的链接)创建这类条目, 并存储在 AR 表中。图 9 所示的 A 与 T 节点之间的关联相应地存储在图 11 的 AR 表中。

每个 T 节点具有其内容(T 节点的名称、T 节点本身的数据、T 节点引用的数据等)。此外, 对于每个 T 节点, 除了存储在 AR 表的各条目中的标识符(ID)之外, 还定义了 T 节点的属性(节点类型(NT); 主题属性)。下面将考虑一个具体实例, 其中, 每个 T 节点仅具有其名称(节点名称(N))作为其内容。

图 12 所示的 T 节点 ID 表的每个条目包括图 9 所示的 T 节点之一的标识符(ID)、为 T 节点定义的属性(节点类型(NT))以及 T 节点的名称(节点名称(N))。

对于图 9 所示的所有 T 节点 T1-1 至 Tn-mn 创建这些条目, 并存储在 T 节点 ID 表中。图 9 所示的所有 T 节点的数据相应地被存储。

图 13 所示的 A 节点 ID 表的每个条目包括图 9 所示的 A 节点之一的标识符(ID)、为 A 节点定义的属性(节点类型(NT'))以及 A 节点的名称(节点名称(N'))。

对于图 9 所示的所有 A 节点 A1 至 An 创建这些条目, 并存储在 A 节点 ID 表中。图 9 所示的所有 A 节点的数据相应地被存储。

要注意, 类似的形式可用来存储 A 与 T 节点之间的关联以及图 9 所示的 A 和 T 节点的数据, 作为表格形式的补充或替代。但是, 在以下描述中, 将考虑采用 AR 表和 ID 表的一个具体实例。

根据 DB 服务器 12 的使用、配置或处理任务, 如图 11 所示, 每个条目可包括 T 节点的内容(节点名称(N))来取代 AR 表中的 T 节点的标识符(ID)。

类似地, 在 AR 表中, 每个条目还可包括 T 节点的内容。

数据搜索

图 14 说明在图 6 和图 7 所示的 DB 服务器 12 中的数据搜索方法。

如图 14 所示, 将考虑一个具体实例, 其中, T 节点 T1 至 Tn 和 T 节点 Tret(T 返回)、(待输出的)搜索结果与某个 A 节点相关联, 在 A 节点与 T 节点 T1 至 Tn 和 Tret 之间定义关联角色 R1 至 Rn 和 Rret, 以及 T 节点 T1 至 Tn 和 Tret 具有节点名称 N1 至 Nn 和 Nret。

在 DB 服务器 12 中, 为 Tret 定义的关联角色 Rret、用于搜索的 A 节点的属性(节点类型 NT; 图 14 中的 ANT1 和 ANT2)、用于搜索的 T 节点的关联角色 R 和节点名称 N 的一个或多个组合用作搜索条件。

如图 14 所示, 例如, 搜索条件由(Rret,(ANT1,ANT2,...),Filter)来表示, 其中 $Filter = ((R1,N1),(R2,N2),...,(Rn,Nn))$ 。

搜索条件还可包含 T 节点的属性 NT(第三条件数据), 稍后进行描述。

在搜索条件中, 包含在 Filter 中的关联角色 R 和 T 节点的节点名称 N 的一个或多个组合(R1,N1)、(R2,N2),...(Rn,Nn)用作搜索的过滤器, 因而以下称作搜索过滤器。

在搜索条件中, A 节点的属性(ANT1,ANT2,...)可省略。

图 15 是第一流程图, 说明在图 6 和图 7 所示的 DB 服务器 12 中的搜索过程的整个过程(S20)。

参照图 15, 在步骤 S200, 例如, DB 服务器 12 接受例如由搜索者采用 PC 102(图 6)或 DB 服务器 12 的输入/输出装置 126 输入的图 14 所示的搜索条件。

在图 16 中详细说明了步骤 S22, 根据稍后进行描述的搜索过滤器来选择关联节点。

在图 17 中详细说明了步骤 S24, 获取稍后进行描述的节点 ID 和节点名称。

在步骤 S202, DB 服务器 12 根据经过 S24 中的处理作为搜索结

果得到的标识符(节点 ID)和 T 节点 Tret 的节点名称(Nret)来创建对搜索者的查询的响应。

对于该响应,仅可使用节点名称 Nret、由节点 Tret 引用的各种数据或者表明节点 Tret 的各种数据。

5 在步骤 S204, 参照图 15, DB 服务器 12 确定搜索者的查询是否已终止。

当搜索者的查询已经终止时, DB 服务器 12 结束此过程, 否则返回到 S200 的处理。

10 图 16 是流程图, 说明基于图 15 所示的搜索过滤器的关关节点选择过程(S22)。

参照图 16, 在接受图 15 所示的 S200 的处理中的搜索条件 (Rret,(ANT1,ANT2,...),Filter), $Filter = ((R1,N1), (R2,N2), ..., (Rn,Nn))$ 时, DB 服务器 12 在步骤 S220 中初始化用于处理的关关节点的列表。

15 在这个关关节点列表中, 在从 AR 表(图 11)得到的 A 节点之间, 在搜索条件中包含 A 节点的属性之一(节点类型; ANT1,ANT2,...)作为其属性(节点类型 NT)的 A 节点的标识符被存储。

要注意, 在从搜索条件中省略 A 节点的属性时 $((ANT1,ANT2,...)=NULL)$, 在 S220 的处理中, 从 AR 表(图 11)所得到的 A 节点的所有标识符被存储。

20 在步骤 S222, DB 服务器 12 确定是否对所有搜索过滤器(R_i, N_i)执行了处理。

如果已经对所有搜索过滤器执行了此处理, 则 DB 服务器 12 进入 S24 的处理(图 15 和图 17)。否则, DB 服务器 12 把仍未处理的搜索过滤器(R_i, N_i)的任一个设置为用于处理的下一个过滤器, 然后进入
25 S224 的处理。

在步骤 S224, DB 服务器 12 搜索 T 节点 ID 表(图 12), 以便检索包含搜索过滤器(R_i, N_i)的节点名称 N_i 的所有条目, 并创建在所检索条目中包含的一组 T 节点标识符(节点 ID 集 T, 其中 $T = \{T_i | \text{节点名称}$

= ID 表中的 N_i })。

5 要注意, 当搜索条件包含 T 节点的属性(节点类型; NT), 并且搜索过滤器由 (R_i, N_i, NT_i) 表示时, 在 S224 的处理中, DB 服务器 12 只需要从 ID 表中检索包含搜索过滤器 (R_i, N_i, NT_i) 的节点名称 N_i 和节点类型 NT_i 的条目, 以及创建在所检索条目中包含的一组 T 节点标识符作为节点 ID 集 T。

在步骤 S226, DB 服务器 12 确定 S224 的处理中得到的节点 ID 集 T 是否为空。

10 当节点 ID 集 T 为空时, DB 服务器 12 执行用于终止搜索过程的处理(向搜索者显示“找到零个匹配”消息等), 然后结束搜索过程。否则, DB 服务器 12 进入 S228 的处理。

在步骤 S228, DB 服务器 12 搜索 AR 表(图 11), 以便更新关联节点列表 A。

15 也就是说, DB 服务器 12 从 AR 表中检索包含搜索过滤器 (R_i, N_i) 的关联角色 R_i 以及在 S224 的处理中得到的节点 ID 集 T 中包含的 T 节点标识符的所有条目, 并把所检索条目中包含的 A 节点标识符存储在关联节点列表 A 中 $(A = \{A_j | \text{角色} = R_i, \text{T 节点标识符} = T_i(\text{所有 } i), \text{A 节点标识符} \in \text{AR 表中的 } A\})$ 。

20 在步骤 S230, DB 服务器 12 确定在 S228 的处理中得到的关联节点列表 A 是否为空。

当关联节点列表 A 为空时, DB 服务器 12 执行用于终止搜索过程的处理, 然后结束搜索过程。否则, DB 服务器 12 进入 S232 的处理。

25 在步骤 S232, DB 服务器 12 读取搜索条件中包含的搜索过滤器, 但不进行处理并返回到 S222 的处理。

图 17 是流程图, 说明图 15 和图 16 所示的节点 ID 和节点名称获取过程 S24。

参照图 17, 在根据搜索过滤器完成关联节点选择过程 S22 时,

在步骤 S240, DB 服务器 12 搜索 AR 表(图 11)以创建 T 节点 ID 集 T。

也就是说, DB 服务器 12 从 AR 表中检索包含搜索条件所包含的关联角色 Rret 以及在 S22(S228)的处理中得到的关关节点列表 A 中包含的 A 节点标识符的所有条目, 并创建所检索条目中包含的一组 T 节点标识符(T 节点 ID 集 T) ($T = \{T_m | \text{角色} = Rret, A \text{ 节点标识符} \in \text{AR 表中的 A}\}$)。

在步骤 S242, DB 服务器 12 确定 S240 的处理中得到的 T 节点 ID 集 T 是否为空。

当 T 节点 ID 集 T 为空时, DB 服务器 12 执行终止处理以结束搜索过程。否则, DB 服务器 12 进入 S244。

在步骤 S244, DB 服务器 12 搜索 T 节点 ID 表(图 12), 以便创建节点 ID 和节点名称集 P。

也就是说, DB 服务器 12 从 ID 表中检索包含在 S240 的处理中创建的 T 节点 ID 集 T 所包含的 T 节点标识符 T_m 中任一个的所有条目, 并创建条目中包含的节点名称 N_m 和 T 节点标识符 T_m 的集合 $P(P = \{(T_m, N_m) | T \text{ 节点标识符} = \text{ID 表中的 } T_m(\text{所有 } m)\})$ 。

在步骤 S246, DB 服务器 12 确定 S244 的处理中得到的 T 节点标识符和 T 节点名称的集合 P 是否为空。

当 T 节点标识符和 T 节点名称的集合 P 为空时, DB 服务器 12 执行终止处理以结束搜索过程。否则, DB 服务器 12 进入 S202 的处理。

这个集合 P 用于在图 15 所示的 S202 的处理中创建对搜索者的响应。

DB 程序 2

图 18 说明在图 6 和图 7 所示的 DB 服务器 12 中运行的 DB 程序 2 的结构。

在图 18 中, 为了清楚地进行说明, 在不需要的情况下省略了表示数据流的线条。

参照图 18, DB 程序 2 包括 DB 管理单元 20、DB 单元 24 和 DB 搜索单元 26。

DB 管理单元 20 包括管理操作接收器 200、AR 条目创建单元 202、ID 条目创建单元 204、AR 数据库管理单元(ARDB 管理单元)206
5 以及 ID 数据库管理单元(IDDB 管理单元)208。

DB 单元 24 包括 AR 数据库(ARDB)240、T 节点 ID 数据库(IDDB)242 和 A 节点 IDDB 244。

DB 搜索单元 26 包括搜索操作接收器 260、搜索条件创建单元 262、搜索控制单元 264、AR 数据库搜索单元(ARDB 搜索单元)266
10 以及 ID 数据库搜索单元(IDDB 搜索单元)268。

例如, DB 程序 2 在记录介质 130(图 7)上携带并从其中读出到 DB 服务器 12, 加载到存储器 124, 并通过具体利用 DB 服务器 12 的硬件在 DB 服务器 12 中的操作系统下运行(以下各程序均类似)。

采用这些组件, DB 程序 2 用来创建以上参照图 9 至图 13 所述的 AR 数据库(图 11)和 ID 数据库(图 12 和图 13), 并采用这些数据库来
15 执行数据搜索(图 14 至 17)。

在 DB 单元 24 中, ARDB 240 存储图 11 所示的 AR 表。

IDDB 242 存储图 12 所示的 T 节点 ID 表。

IDDB 244 存储图 13 所示的 A 节点 ID 表。

图 18 说明一个具体实例, 其中, 图 12 和图 13 所示的 T 节点 ID 表和 A 节点 ID 表存储在 IDDB 242 和 244 中。但是, T 节点 ID 表和 A 节点 ID 表可存储在相同数据库中。
20

T 节点 ID 表和 A 节点 ID 表不需要总是分开创建, 而是它们可共同创建一个数据库中。

在 DB 管理单元 20 中, 管理操作接收器 200 从输入/输出装置 126(图 7)或者通过网络 100 从 PC 102(图 6)接收管理或修改 AR 表和 ID 表中存储的数据的操作, 并把操作输出到 ARDB 管理单元 206 和 IDDB 管理单元 208。
25

管理操作接收器 200 接收用于指定 A 节点和 T 节点、A 节点与 T 节点之间的关联、A 节点与 T 节点之间定义的关联角色 R(链接)、分配给 A 节点和 T 节点的标识符(ID)、分配给 T 节点的节点名称(N)以及为 T 节点定义的属性(图 9)的用户操作, 并把操作输出到 AR 条目创建单元 202 和 ID 条目创建单元 204。

例如, 管理操作接收器 200 在输入/输出装置 126 上显示表示 A 节点和 T 节点、如图 14 所示的其间关联的用户界面(UI)图像, 接收 UI 图像上的用户操作, 以及接受指定。

AR 条目创建单元 202 根据从管理操作接收器 200 输入的用户指定来创建图 11 所示的 AR 表的条目, 并把条目输出到 ARDB 管理单元 206。

ARDB 管理单元 206 把从 AR 条目创建单元 202 输入的 AR 表的条目添加到 ARDB 240 存储的 AR 表中。

ARDB 管理单元 206 根据从管理操作接收器 200 输入的用户操作来修改 ARDB 240 中存储的 AR 表的内容。

ARDB 管理单元 206 根据 ARDB 搜索单元的搜索请求来检索 ARDB 240 中存储的 AR 表的条目, 并把条目输出到 ARDB 搜索单元 266。

ID 条目创建单元 204 根据从管理操作接收器 200 输入的用户指定来创建图 12 和图 13 所示的 T 节点和 A 节点 ID 表的条目, 并把条目输出到 IDDB 管理单元 208。

IDDB 管理单元 208 把从 ID 条目创建单元 204 输入的 T 节点 ID 表的条目添加到 IDDB 242 存储的 T 节点 ID 表中。

IDDB 管理单元 208 把从 ID 条目创建单元 204 输入的 A 节点 ID 表的条目添加到 IDDB 244 存储的 A 节点 ID 表中。

IDDB 管理单元 208 根据从管理操作接收器 200 输入的用户操作来修改 IDDB 242 和 244 中存储的 ID 表的内容。

IDDB 管理单元 208 根据 IDDB 搜索单元 268 的搜索请求来检索

IDDB 242 和 244 中存储的 ID 表的条目, 并把条目输出到 IDDB 搜索单元 268。

在 DB 搜索单元 26 中, 搜索操作接收器 260 从输入/输出装置 126(图 7)或者通过网络 100 从 PC 102(图 6)接收指定用于图 14 至 17 所示的搜索过程的搜索条件(图 14, 还可包含可选 T 节点的属性(节点类型(NT)))的搜索者操作。

搜索操作接收器 260 把所接收操作输出到搜索条件创建单元 262。

例如, 当搜索操作接收器 260 接受自然语言查询形式的搜索条件时, 搜索条件创建单元 262 分析查询语句以便提取单词。

随后, 搜索条件创建单元 262 通过 ARDB 搜索单元 266、IDDB 搜索单元 268、ARDB 管理单元 206 以及 IDDB 管理单元 208 来搜索 ARDB 240 和 IDDB 242、244 中存储的 AR 表和 ID 表, 并提取用作搜索条件的单词。

此外, 搜索条件创建单元 262 按照查询语句的结构组合所提取单词, 检索图 14 所示的 $(Rret, (ANT1, ANT2, \dots), ((R1, N1), (R2, N2), \dots, (Rn, Nn)))$ 形式的搜索条件, 并把搜索条件输出到搜索控制单元 264。

要注意, 当搜索者直接指定图 14 所示的 $(Rret, (ANT1, ANT2, \dots), ((R1, N1), (R2, N2), \dots, (Rn, Nn)))$ 形式的搜索条件时, 搜索条件创建单元 262 可省略。

搜索条件创建单元 262 可以是用于帮助搜索者检索搜索条件 $(Rret, (ANT1, ANT2, \dots), ((R1, N1), (R2, N2), \dots, (Rn, Nn)))$ 的工具。

搜索控制单元 264 根据从搜索条件创建单元 262(搜索操作接收器 260)输入的搜索条件 $(Rret, (ANT1, ANT2, \dots), ((R1, N1), (R2, N2), \dots, (Rn, Nn)))$ 控制 ARDB 搜索单元 266 和 IDDB 搜索单元 268, 以便通过 ARDB 管理单元 206 和 IDDB 搜索单元 208 来执行在 ARDB 240(AR 表; 图 11)和 IDDB 242、244(ID 表; 图 12 和图

13)中的搜索,如图 15 至 17 所示。

当通过基于搜索条件的搜索得到搜索结果(集合 P; 图 17)时,搜索控制单元 264 根据搜索结果创建响应,在输入/输出装置 126(图 7)中显示此响应,或者通过网络 100(图 6)在 PC 102 的输入/输出装置 126 中向搜索者显示此响应。

ARDB 搜索单元 266 在搜索控制单元 264 的控制下通过 ARDB 管理单元 206 搜索 ARDB 240(AR 表; 图 11),并把搜索结果返回给搜索控制单元 264。

IDDB 搜索单元 268 在搜索控制单元 264 的控制下通过 IDDB 管理单元 208 搜索 IDDB 242、244(ID 表; 图 12 和图 13),并把搜索结果返回给搜索控制单元 264。

整体操作

下面将针对具体实例来描述图 6 和图 7 所示的 DB 服务器 12(DB 程序 2; 图 18)的整体操作。

AR 表和 ID 表的创建

首先描述通过 DB 程序 2 的 DB 管理单元 20 创建 AR 表和 ID 表的过程。

图 19 说明对图 6 和图 7 所示的 DB 服务器 12(DB 程序 2; 图 18)的数据输入以及对其中包含的数据执行搜索的搜索条件。

例如,如图 19 所示相关的数据被输入 DB 程序 2 的管理操作接收器 200。

图 19 所示的数据包含如以下所述相关的 A 节点和 T 节点,以及节点名称被分配给 T 节点(但是, A 节点的属性(节点类型(NT)和节点名称,以及 T 节点的属性(节点类型(NT))从以下小节(1)至(8)以及图 19 中省略)。

(1) “A 节点 A1” 和 “T 节点 T11” 相关在一起,在它们之间定义关联角色 R “作者”,以及节点名称 “莎士比亚”(具有相同名称的不同人)被分配给 “T 节点 T11”。

(2) “A 节点 A9” 和 “T 节点 T92、T41” 相关在一起，在它们之间定义关联角色 R “作品” 和 “作者”，以及节点名称 “威尼斯商人” 和 “莎士比亚” 被分配给 “T 节点 T92 和 T42”。

5 (3) “A 节点 A4” 和 “T 节点 T41、T42” 相关在一起，在它们之间定义关联角色 R “作者” 和 “作品”，以及节点名称 “哈姆雷特” 被分配给 “T 节点 T42”。

(4) “A 节点 A13” 和 “T 节点 T42” 相关在一起，以及在它们之间定义关联角色 R “剧本”。

10 (5) “A 节点 A19” 和 “T 节点 T42” 相关在一起，以及在它们之间定义关联角色 R “原著”。

(6) “A 节点 A10” 和 “T 节点 T42” 相关在一起，以及在它们之间定义关联角色 R “原著”。

15 (7) “A 节点 A10” 和 “T 节点 T103、T101” 相关在一起，在它们之间定义关联角色 R “出版” 和 “译本”，以及节点名称 “OO 出版社” 被分配给 “T 节点 T103”。

(8) “A 节点 A19” 和 “T 节点 T191” 相关在一起，以及在它们之间定义关联角色 R “译本”。

管理操作接收器 200 接收输入数据，并把数据输出到 AR 条目创建单元 202 和 ID 条目创建单元 204。

20 AR 条目创建单元 202 从图 19 所示的数据中创建 AR 表的每个条目，并把条目输出到 ARDB 管理单元 206。

图 20 说明由 AR 条目创建单元 202(图 18)和 ARDB 管理单元 206 创建并存储在 ARDB 240 中的 AR 表。

25 要注意，在以下附图中，NULL(“空”)表明没有任何属性(节点类型)/名称(节点名称)。

ARDB 管理单元 206 把从 AR 条目创建单元 202 输入的 AR 表的条目依次添加到 ARDB 240 存储的 AR 表中。

作为通过 AR 条目创建单元 202 和 ARDB 管理单元 206 的处理的

结果，图 20 所示的 AR 表从图 19 所示的数据中创建并存储在 ARDB 240 中。

图 21 说明由 ID 条目创建单元 204(图 18)和 IDDB 管理单元 208 创建并存储在 IDDB 242 中的 T 节点 ID 表。

5 ID 条目创建单元 204 从图 19 所示的数据中创建 T 节点 ID 表的条目，并把条目输出到 IDDB 管理单元 208。

IDDB 管理单元 208 把从 ID 条目创建单元 204 输入的 T 节点 ID 表的条目依次添加到 IDDB 242 存储的 ID 表中。

10 作为通过 ID 条目创建单元 204 和 IDDB 管理单元 208 的处理的结果，图 21 所示的 T 节点 ID 表从图 19 所示的数据中创建并存储在 IDDB 242 中。

图 22 说明由 ID 条目创建单元 204(图 18)和 IDDB 管理单元 208 创建的 A 节点 ID 表。

15 ID 条目创建单元 204 从图 19 所示的数据中创建 A 节点 ID 表的条目，并把条目输出到 IDDB 管理单元 208。

IDDB 管理单元 208 把从 ID 条目创建单元 204 输入的 ID 表的条目依次添加到 IDDB 244 存储的 A 节点 IDDB 表中。

20 作为通过 ID 条目创建单元 204 和 IDDB 管理单元 208 的处理的结果，与图 22 所示类似的 A 节点 ID 表从图 19 所示的数据中创建并存储在 IDDB 244 中。

数据搜索

例如，当搜索者以查询语句的形式向 DB 服务器 12 的输入/输出装置 126(图 7)输入搜索条件“出版剧作家莎士比亚作为作者写作的作品之一 - 戏剧哈姆雷特的原著的译本的出版商的名称是什么？”
25 时，搜索条件创建单元 262 分析这个查询语句，并把查询语句分析为两个部分，即图 19 所示的第一半和第二半。

在查询语句的第二半“剧作家莎士比亚作为作者写作的作品”中，为了检索与“作品”有关的数据，搜索条件创建单元 262 把要作

为搜索结果的 T 节点 Tret 的关联角色 Rret 设置为“作品”。

搜索条件创建单元 262 从第二半“莎士比亚作为作者写作”中创建搜索过滤器(R1=“作者”，N1=“莎士比亚”)。

5 此外,搜索条件创建单元 262 从 T 节点 Tret 的关联角色(Rret=“作品”)和搜索过滤器(R1=“作者”，N1=“莎士比亚”)创建与查询语句的第二半对应的搜索条件(Rret,(ANT1),((R1,N1)))=(作品,(空),((作者,莎士比亚)))。

10 在查询语句的第一半“出版戏剧哈姆雷特的原著的译本的出版商的名称是什么?”中,为了检索与“出版的出版商”有关的信息,搜索条件创建单元 262 把要作为搜索结果的 T 节点 Tret 的关联角色 Rret 设置为“出版”。

15 搜索条件创建单元 262 从第一半包含戏剧“哈姆雷特”的“原著”的条件创建第一搜索过滤器(R1=“原著”，N1=“哈姆雷特”)，以及从第一半包含的“译本”创建第二搜索过滤器(R2=“译本”，N2=“空(未指定)”)。

20 此外,搜索条件创建单元 262 从关联角色(Rret=“出版”)、第一搜索过滤器(R1=“原著”，N1=“哈姆雷特”)和第二过滤器(R2=“译本”，N2=“空”)创建与查询语句的第一半对应的搜索条件(Rret,(ANT1),((R1,N1),(R2,N2)))=(出版,(空),((原著,哈姆雷特),(译本,空)))。

根据搜索条件创建单元 262 创建的搜索条件,搜索控制单元 264 通过 ARDB 搜索单元 266 和 IDDB 搜索单元 268 来搜索 ARDB 240(AR 表;图 20)和 IDDB 242(T 节点 ID 表;图 21),以便获得搜索结果,如以下所述。

25 首先,由从查询语句的第二半得到的搜索条件,搜索控制单元 264 执行以下操作。

(1) 搜索控制单元 264 参考 IDDB 242(图 18)中存储的 T 节点 ID 表以检索其节点名称为“莎士比亚”的 T 节点的所有标识符(ID),并

获得 T11、T31、T41、T51 和 T81(图 21)作为这个处理的结果。

5 (2) 搜索控制单元 264 参考 ARDB 240 中存储的 AR 表(图 20)以检索其中角色为“作者”以及 T 节点标识符匹配(1)的处理中得到的 T 节点标识符的所有 A 节点标识符, 并且获得 A1、A4 和 A9 作为这个处理的结果。

(3) 搜索控制单元 264 参考 AR 表以在通过(2)的处理得到的 A 节点标识符之中检索与其关联角色 R 为“作品”的 A 节点对应的 T 节点标识符(ID; 一般为复数), 并获得 T42 和 T92 作为这个处理的结果。

10 (4) 搜索控制单元 264 参考 ID 表, 并把与通过(3)的处理得到的 T 节点标识符对应的标识符的节点名称与 T 节点标识符(ID)设置在一起作为搜索结果。

也就是说, 搜索控制单元 264 根据通过(1)至(4)的处理从查询语句的第二半得到的搜索条件(作品, (空), ((作者, 莎士比亚)))来执行搜索, 并获得搜索结果(T42, 哈姆雷特)以及(T92, 威尼斯商人)。

15 随后, 根据从查询语句的第一半得到的搜索条件以及与第二半对应的搜索结果, 搜索控制单元 264 执行以下操作。

(5) 从对应于查询语句的第二半的搜索结果(T42, 哈姆雷特)和(T92, 威尼斯商人), 搜索控制单元 264 选择其节点名称对应于第一搜索过滤器(原著, 哈姆雷特)的(T42, 哈姆雷特), 并获得其节点标识符(ID)T42。

(6) 搜索控制单元 264 参考 AR 表以检索其中关联角色为“原著”以及 T 节点标识符匹配通过(5)的处理得到的节点标识符(ID)的所有 A 节点标识符, 并且获得 A10 和 A19 作为结果。

25 (7) 搜索控制单元 264 参考 AR 表以在通过(6)的处理得到的 A 节点标识符之中检索其角色为“译本”的所有标识符, 并且获得 A10 和 A19 作为结果。

(8) 搜索控制单元 264 参考 AR 表以在通过(7)的处理得到的 A 节点标识符之中检索与其角色为“出版”的 A 节点对应的 T 节点标识

符,并获得 T103 作为结果(当存在来自多个出版商的译本时,获得多个 T 节点)。

5 (9) 搜索控制单元 264 参考 ID 表,并把与通过(8)的处理得到的 T 节点标识符对应的节点 ID 的节点名称与节点标识符设置在一起作为搜索结果(T103, OO 出版社)。

也就是说,搜索控制单元 264 根据通过(5)至(9)的处理从查询语句的第一半得到的搜索条件(出版,(空),((原著,哈姆雷特),(译本,空)))来执行搜索,并获得搜索结果(T013, OO 出版社)。

10 (10) 搜索控制单元 264 在输入/输出装置 126(图 7)等中向搜索者显示搜索结果。

第二数据库系统 3

图 23 说明根据本发明的一个实施例的第二 DB 系统 3 的配置。

15 参照图 23,第二 DB 系统 3 通过经由网络 100 互连 DB 服务器 12、用于在与 DB 服务器 12 相似的硬件(图 7)上操作 DB 程序 2 的 DB 管理单元 20 和 DB 单元 24 的 DB 服务器 30、以及用于在与 DB 服务器 12 相似的硬件上操作 DB 程序 2 的 DB 搜索单元 26 的检索装置 32 来配置。

这样,DB 程序 2 不需要始终在一台计算机上运行,而是可分布到通过网络互连的多台计算机上运行。

20 第三 DB 系统 4

下面将描述配置成根据目录结构存储和搜索关联信息的本发明的一个实施例的第三 DB 系统 4。

图 24 说明第三 DB 系统 4 的配置。

25 参照图 24,DB 系统 4 通过经由网络 100 互连 PC 102、检索装置 40、DB 管理服务器 5 以及 n 个 DB 服务器 6-1 至 6-n 来配置。

要注意,当未指定诸如 DB 服务器 6-1 至 6-n 等的多个组件的任一个时,它可简称作 DB 服务器 6。

检索装置 40、DB 管理服务器 5 和 DB 服务器 6 的硬件采用图 7

所示的配置。

当 DB 服务器 6 包括象在图 6 所示的第一 DB 系统 1 的 DB 服务器 12(图 6)和 DB 程序 2(图 18)的情况下那样的搜索功能时,检索装置 40 可变成不需要的。

5 此外,在 DB 系统 4 中,可提供复制服务器(镜像服务器;DB 服务器 6'),与每个 DB 服务器 6 对应。

DB 服务器 6-1 至 6-n 以下可称作 DB 服务器 A 至 N。

接下来将描述图 24 的 DB 系统 4 中的数据表示。

图 25 说明图 24 的 DB 系统 4 中存储的数据的图形表示。

10 图 26 说明图 25 所示的数据的目录信息树表示。

如以上参照图 8 所述,当为 A 和 T 节点定义了节点名称等(A 节点的节点名称等为“与关联节点对应的数据”,以及 T 节点的节点名称等为“与主题节点对应的数据”),并且在 A 与 T 节点之间定义了关联角色时,通过这些节点相关的相关数据可通过如图 25 所示的图形形式来表示。

15 图 25 说明一种情况,其中,“位置信息”、“商店”、“区域”被定义为节点(T1, T3-1 和 T3-2)的节点名称,用于定义 T 和 A 节点的节点类型(类),“商店 1”和“区域 3”被定义为 T 节点(T2-1 和 T2-2)的节点名称,“位置”被定义为 A 节点(A1)的节点名称,以及“建筑物”和“地点”被定义为 T 和 A 节点之间的关联角色。

A 节点 A2-1 和 A2-2 表明用于定义节点类型(类)的节点以及作为其示例的节点(与实体对应的节点)。这样,节点 A2-1 和 A2-2 是特殊节点,因为它们实现不是基于由用户根据目的独立定义的角色、而是基于由系统预定义的两个角色“类”和“示例”的 T 节点的关联。

25 图 25 所示的图形表示的相关数据可通过 DIT(目录信息树)的形式来表示,它是图 26 所示的 LDAP(轻型目录访问协议)的数据模型。

也就是说,根据 DIT 表示,图 25 所示的相关数据可由以下条目来表示:对应于 A 节点的条目以及对应于关联角色且具有 T 节点数

据作为属性的条目，如图 26 中的实线所示，它们设置在作为各 DB 服务器 6 的顶部条目虚拟提供的虚拟根条目之下；以及表示第一、第二、...关联(关联 # 1、# 2...)的条目，如图 26 中的虚线所示。

换言之，在图 26 所示的情况中，与两个角色“建筑物”和“地点”对应的条目直接在表明节点类型(assocType)为“位置信息”以及用于充当角色的成员“商店 1”和“区域 3”为相应条目的属性的相关数据的条目“位置”之下提供。

要注意，在图 26 所示的 DIT 形式中，“目录”被定义成以与现实相关的易于理解形式来构造与现实中的各种对象有关的数据，以便于数据的定位以及提供用于检索和更新数据的方式。

“条目”被定义以便根据对象类对与现实的对象有关的数据排列/分类以及把数据表示为目录信息，或者被定义为目录中存储的数据。

“目录信息树”被定义为在以分层方式管理条目(目录信息)时表示树结构中的分层关系。

换言之，图 26 中的正方形对应于条目，并且它是在树结构中表示其分层关系的目录信息树(DIT)。

如上所述，条目是与对象有关的一组数据，以及与这个对象有关的数据称作“属性”。

属性包括“属性类型”和一个或多个值“属性值”。参照图 26，每个条目结构以“属性类型：属性值”的形式表示为 LDAP 中的条目表示，如下表 21 至 23 所示。

在紧接“ou=关联 # 1”之下的条目中，条目的各属性被定义，其方式例如是，属性类型对象类的属性值为关关节点，属性类型 cn(普通名称)的属性值为位置、....

表 21

dn:associd=关关节点 ID

对象类：关关节点

5

associd: 关联节点 ID
assocType: 位置信息
cn: 位置
角色: 建筑物
角色: 地点

表 22

10

dn:角色 = 建筑物, associd=关联节点 ID
对象类: 关联角色
cn: 建筑物
角色: 建筑物
成员: 商店 1

表 23

15

dn:角色 = 地点, associd=关联节点 ID
对象类: 关联角色
cn: 地点
角色: 地点
成员: 区域 3

图 27 说明分割以目录结构表示的相关数据的方法。

20

图 28 说明图 27 所示的 DB 服务器 6-1(A)的条目 dn_A2 与 DB 服务器 6-n(N)的条目 dn_N 之间的引用关系。

如图 27 所示，以 DIT 形式表示的相关数据可根据目录结构来分割。

25

在图 27 所示的情况中，相关数据由包含条目 dn_A 以及下级条目 dn-A1 和 dn_A2 的目录结构来表示，与条目 dn_A1 或下级条目的目录对应的相关数据被分割，对应于条目 dn_B 至 dn_(N-1)，以及与条目 dn_A2 或下级条目的目录对应的相关数据对应于条目 dn_N 或下级条目的目录结构(要注意，在图 27 所示的实例中，具有条目 dn_B 至 dn_N 作为顶部条目的目录结构为子树，以及包含这类子树以及条

目 dn_A 作为顶部条目的目录结构为目录树)。

例如,如图 28 所示,图 27 所示的 DB 服务器 6-n(N)的顶部条目 dn_N 在 DB 服务器 6-1(A)的条目 dn_A2 之下,并且 dn_N 引用 dn_A2。

5 因此,按照目录树分割的相关数据可经过分割并存储在图 27 所示的 DB 服务器 6-1 至 6-n 中。

图 29 说明用于管理经过分割并存储在图 27 所示的 DB 服务器 6-1 至 6-n(DB 服务器 A 至 N)中的相关数据的目录树表。

为了管理经过分割并存储在图 27 和图 28 所示的 DB 服务器 6-1 至 6-n(A 至 N)中的相关数据,使用与图 29 所示相似的目录树表。

10 参照图 29,目录树表包含 DB 服务器 6-1 至 6-n 的服务器名称(A 至 N)、DB 服务器 6 的顶部条目(dn_A 至 dn_N; 与图 26 的虚拟条目相似)、由 DB 服务器 6 的顶部条目引用的条目(引用条目)、以及在 DB 服务器 6 中以对应方式存在复制服务器(镜像服务器(第二数据库服务器))时的装置名称(A'、C'等)。

15 图 30 说明用于把图 27 和图 28 所示的 DB 服务器 6-1 至 6-n 中存储的目录树(或子树)的顶部条目 dn_A 至 dn_N 与用于对目录树(或子树)中存储的相关数据分类的分类类型/分类值(分类数据)相关的数据表。

20 如图 25 和图 26 所示,以 DIT 形式的目录结构所表示、按照目录结构分割并存储在 DB 服务器 6-1 至 6-n(A 至 N)中的相关数据可看作是分别在 DB 服务器 6-1 至 6-n(A 至 N)中具有公共关联的数据集。

因此,可对于图 27 和图 28 所示的 DB 服务器 6-1 至 6-n(A 至 N)中存储的目录树(或子树)定义表示用于对所存储相关数据分类的属性等的分类类型和分类值。

25 为了管理 DB 服务器 6-1 至 6-n(A 至 N)中存储的相关数据,如图 30 所示,采用一种数据表,它包含为顶部条目 dn_A 至 dn_N 之下的子树 a 至 n 所定义的分类类型和分类值、顶部条目(dn_A 至 dn_N)以及以对应方式示于图 27 和图 28 的 DB 服务器 6-1 至 6-n(A 至 N)的

DB 服务器 6-1 至 6-n(A 至 N)的顶部条目或下级条目的子树。

要注意, 图 30 所示的“分类类型”对应于关联类型(A 节点(图 22)的节点类型; 属性)。

“分类值”对应于 LDAP 数据模型中的示例。

5 可能是图 27 至 30 所示的顶部条目的条目的具体实例包括但不限于:

(1)包含某个分类类型(关联类型; A 节点的节点类型)的具体分类值(示例)的子树的根条目; 以及

10 (2)包括其中含某个具体分类值(示例)的相关数据的子树的根条目。

在这里, 描述了其中“分类类型”对应于关联类型以及“分类值”对应于其示例的情况。但是, 这只是图 30 与图 27 和图 28 相关的一个实例。“分类类型”可能不对应于关联类型, 以及“分类值”可能不对应于其示例。

15 例如, 图 30 所示的子树包含“分类类型: 类别, 以及类型值: 附属”的相关数据。这是其中“分类类型”不对应于关联类型以及“分类值”不对应于其示例的一个实例。

例如, 表示为包含图 30 所示的数据表的顶部条目 dn_B 的条目分类类型的“位置信息”是(1)“包含某个分类类型(关联类型; A 节点的节点类型)的具体分类值(示例)的子树的根条目”的分类类型的一个实例, 它包含分类值“位置(位于)”作为分类值之一。

这个条目表明具有“位置(位于)”作为分类值的所有相关数据均存储在 DB 服务器 6-2(B)中。

25 作为分类类型“位置信息”的分类值的其它实例, 可列举分类值“相邻(紧接)”等。

或者, 写为包含图 30 所示的顶部条目 dn_(N-1)的条目分类类型的“建筑物”是(2)“包括其中含某个具体分类值(示例)的相关数据的子树的根条目”的分类类型的一个实例, 它包括作为分类值之一的分

类值“商店 1”。

这个条目表明具有“商店 1”作为分类值的所有相关数据均存储在 DB 服务器 6-(n-1)(N-1)中。

相关数据的创建和搜索

5 下面将描述采取以上参照图 25 至 30 所述的 DIT 形式的目录结构来表示的相关数据的创建以及对已创建相关数据的搜索。

第一 DB 管理程序 50

图 31 说明在图 24 所示的 DB 管理服务器 5 中运行的第一 DB 管理程序 50 的结构。

10 参照图 31, DB 管理程序 50 包括用户界面(UI)单元 500、数据登记单元 502、数据传送单元 504、相关数据创建及管理单元 510(目录树创建部件和数据相关部件)、树信息提供单元 512、目录树表创建单元 522(顶部条目相关部件)、目录树表管理单元 524、目录树表 DB 526、数据表创建单元 532(分类数据相关部件)、数据表管理单元 534
15 以及数据表 DB 536。

采用这些组件, DB 管理程序 50 创建以图 26 所示的 DIT 形式的目录结构所表示的相关数据。

DB 管理程序 50 分割已创建相关数据并将其存储在如图 27 和图 28 所示的 DB 服务器 6-1 至 6-n 中, 并且通过采用图 29 和图 30 所示
20 的目录树表和数据表来管理相关数据。

另外, DB 管理程序 50 适当地提供目录树表和数据表中包含的数据(树数据), 它是对检索装置 40(图 24)搜索所需的。

图 32 说明图 31 所示的 DB 管理程序 50 向 DB 管理服务器 5 的输入/输出装置 126(图 7)显示的 GUI 图像。

25 在 DB 管理程序 50 中, UI 单元 500 为 DB 管理服务器 5 或 PC 102 的用户提供 GUI 环境, 并向输入/输出装置 126(图 7)显示与图 32 所示类似的 GUI 图像。

UI 单元 500 接受所显示 GUI 图像上登记、管理或修改相关数据

的用户操作，把操作输出到 DB 管理程序 50 的组件、如数据登记单元 502 和相关数据创建及管理单元 510，以及控制其操作。

另外，UI 单元 500 向输入/输出装置 126 显示/输出由 DB 管理程序 50 的各组件所创建的数据。

5 要注意，UI 单元 500 可安装在 DB 管理服务器 5 或 PC 102 中，以及功能可提供给 PC 102 上的用户。

相关数据创建及管理单元 510 控制目录树表创建单元 522 和数据表创建单元 532 等的操作，由从 UI 单元 500 输入的表明 T 和 A 节点及关联属性的数据创建以图 26 所示的 DIT 形式的目录结构所表示的相关数据，以及通过采用图 29 和图 30 所示的目录树表和数据表来管理已创建相关数据。

对请求作出响应，相关数据创建及管理单元 510 向树数据提供单元 512 输出目录树表和数据表所包含的用于检索装置 40 的检索的数据(树数据)。

15 另外，根据用户操作，相关数据创建及管理单元 510 分割以图 27 和图 28 所示的 DIT 形式的目录结构所表示的相关数据，并把相关数据输出到数据登记单元 502。

数据登记单元 502 通过数据传送单元 504 把从相关数据创建及管理单元 510 输入的相关数据传送给 DB 服务器 6-1 至 6-n，并请求对相关数据的登记。

树信息提供单元 512 响应来自检索装置 40 的请求而提供从相关数据创建及管理单元 510 输入的树数据。

目录树表创建单元 522 在相关数据创建及管理单元 510 的控制下创建图 29 所示的目录树表，并把该表输出到目录树表管理单元 524。

25 目录树表管理单元 524 把从目录树表创建单元 522 输入的目录树表存储在目录树表 DB 526 中。

对请求作出响应，目录树表 DB 526 把目录树表 DB 526 中存储的目录树表输出到相关数据创建及管理单元 510。

要注意，虽然在图 18 所示的 DB 程序 2 中 DB 管理单元 20 执行从相关数据创建 AR 表和 ID 表的处理，但是在图 31 所示的 DB 管理程序 50 中，相关数据创建及管理单元 510 创建目录结构的相关数据(图 21 和图 22)，以及 DB 服务器 6 通过相关数据创建及管理单元 510 和数据传送单元 504 接收已创建的目录结构的相关数据，并存储/管理相关数据。

数据表创建单元 532 在相关数据创建及管理单元 510 的控制下创建图 30 所示的数据表，并把数据表输出到数据表管理单元 534。

数据表管理单元 534 把从数据表创建单元 532 输入的数据表存储在数据表 DB 536 中。

另外，对请求作出响应，数据表管理单元 534 把数据表 DB 536 中存储的数据表输出到相关数据创建及管理单元 510。

下面将描述 DB 管理程序 50 的整个过程。

图 33 是流程图，说明通过使用图 31 所示的 DB 管理程序 50 的相关数据的登记过程 S30。

参照图 33，在步骤 S300，DB 管理程序 50 或 PC 102 的 UI 单元 500 向输入/输出装置 126 显示图 32 所示的 GUI 图像。

用户在所显示的 GUI 图像上进行操作，并向各字段输入与新增相关数据有关的数据。

在步骤 S302，相关数据创建及管理单元 510 根据在步骤 S300 的处理中输入的相关数据来标识关联(题目；例如“莎士比亚的作品”)。

在步骤 S304，相关数据创建及管理单元 510 标识相关数据的节点类型(分类类型；例如“出处信息”)，并把这个类型设置为条目属性。

在步骤 S306，相关数据创建及管理单元 510 通过数据表管理单元 534 参考数据表，并根据在步骤 S302 和 S304 的处理中标识的题目和节点类型来确定把已创建条目分类到的子树。

在步骤 S310，相关数据创建及管理单元 510 通过目录树表管理

单元 524 参考目录树表, 并确定在其中存储步骤 S306 的处理中得到的子树的 DB 服务器 6。

在步骤 S312, 相关数据创建及管理单元 510 根据与通过 UI 单元 500 输入的相关数据有关的数据来创建相应条目。

5 在步骤 S314, 相关数据创建及管理单元 510 通过数据登记单元 502 和数据传送单元 504 把添加的相关数据条目传送到在步骤 S310 的处理中确定的 DB 服务器 6。DB 服务器 6 判断所接收相关数据条目是否已被存储。

10 要注意, 相关数据创建及管理单元 510 在对 DB 服务器 6 中的存储预处理时采用 DB 服务器 6 中的数据。

每个条目 dn(区别名称)被分配给这个数据, 以及 DIT 中的所有条目可由 dn 唯一标识。因此, 可判断是否存在相同条目。

DB 管理程序 50 在接收到来自 DB 服务器 6 的已经存储所添加相关数据条目的结果时结束此过程, 否则进入步骤 S316 的处理。

15 在步骤 S316, 相关数据创建及管理单元 510 在所确定的 DB 服务器 6 中登记所添加相关数据条目。

图 34 是流程图, 说明通过图 31 所示的 DB 管理程序 50 对数据表(图 30)的更新过程 S34。

20 在步骤 S340, 用户操作 DB 管理服务器 5(图 24)或 PC 102 来指定要改变的子树, 并设置指定子树的分类类型/分类值。

相关数据创建及管理单元 510 通过 UI 单元 500 接收操作。

在步骤 S342, 用户操作 DB 管理服务器 5(图 24)或 PC 102 来设置指定子树的顶部条目。

相关数据创建及管理单元 510 通过 UI 单元 500 接收操作。

25 在步骤 S344, 相关数据创建及管理单元 510 判断在 S340 和 S342 的处理中接收的操作所表明的分类类型、分类值和顶部条目是否都已经对指定子树设置。

相关数据创建及管理单元 510 通过数据表管理单元 534 读取数据

表，并在已经对指定子树设置了所有数据时结束此过程，否则进入 S346 的处理。

在步骤 S346，相关数据创建及管理单元 510 把所接收的设定值 (指定子树、分类类型、分类值和顶部条目) 输出到数据表管理单元 534。

数据表管理单元 534 把从相关数据创建及管理单元 510 输入的设定值设置到数据表中，并将其存储在数据表 DB 536 中。

图 35 是流程图，说明通过图 31 所示的 DB 管理程序 50 对目录树表(图 29)的更新过程 S36。

参照图 35，在步骤 S360，UI 单元 500 从 DB 管理服务器 5 接收分割目录信息树的用户操作。

在步骤 S362，相关数据创建及管理单元 510 从目录树表管理单元 524 获得目录树表(图 29)，检索表明 DB 服务器 6 的配置和目录结构的数据，并向用户显示数据。

在步骤 S364，UI 单元 500 接收在按照对所显示的目录结构的用户操作进行分割之后的子树的顶部条目和引用条目的设定。

在步骤 S366，相关数据创建及管理单元 510 根据分割目录信息树的用户操作来判断是否设置在其中存储已分割子树的 DB 服务器 6 的复制服务器。

DB 管理程序 50 在设置复制服务器时进入 S368 的处理，否则进入 S370 的处理。

在步骤 S368，相关数据创建及管理单元 510 当在 S366 的处理中配置了复制服务器的设定时确定复制服务器。

在步骤 S370，当分割之后的子树的顶部条目和引用条目和复制服务器被设置时，相关数据创建及管理单元 510 把复制服务器的名称输出到目录树表管理单元 524。

在写入目录树表 DB 526 的指定部分时，目录树表管理单元 524 反映从相关数据创建及管理单元 510 输入的数据。

在步骤 S372, 相关数据创建及管理单元 510 判断是否已经对所有 DB 服务器 6 完成目录树表的更新过程。

当还没有对所有 DB 服务器 6 完成更新过程时, DB 管理程序 50 返回到 S364 的处理。

5 随后将描述 DB 管理服务器 5 的整个过程。

图 36 是顺序图, 说明图 31 所示的 DB 管理程序 50 的整个操作 S40。

参照图 36, 在步骤 S400, 用户把相关数据输入 DB 管理程序 50 的 UI 单元 500。

10 在步骤 S402, UI 单元 500 把与输入的相关数据有关的数据输出到相关数据创建及管理单元 510。

相关数据创建及管理单元 510 接收数据。

15 在步骤 S404, 相关数据创建及管理单元 510 向数据表管理单元 534 输出用于对与所接收相关数据有关的数据之中的相关数据、如题目和节点类型进行分类的分类数据(分类类型/分类值)。

在步骤 S406, 数据表管理单元 534 从相关数据创建及管理单元 510 接收用于对相关数据分类的数据, 并把相应子树的顶部条目返回给相关数据创建及管理单元 510。

相关数据创建及管理单元 510 接收子树的顶部条目。

20 要注意, 当没有相应子树时, DB 服务器 6-1 的顶部条目(即根条目)被返回给相关数据创建及管理单元 510 来替代相应子树的顶部条目。

25 在 S406 的处理中, 采用与数据表(图 30)中存储的分类类型和分类值对应的数据, 以及与通过图 32 所示的 GUI 图像输入的相关数据有关的数据都可由相关数据创建及管理单元 510 使用。

也就是说, 相关数据创建及管理单元 510 把节点类型和特定示例的组合提供给数据表管理单元 534, 从而数据表管理单元 534 可获得与用于对相关数据分类的数据对应的子树的顶部条目。另外, 例如,

作为分类类型的“类别”以及作为分类值的“莎士比亚的作品”可提供给数据表管理单元 534。

在步骤 S408, 相关数据创建及管理单元 510 把在 S406 的处理中得到的子树的顶部条目输出到目录树表管理单元 524。

5 目录树表管理单元 524 接收子树的顶部条目。

在步骤 S410, 目录树表管理单元 524 通过采用子树的所接收顶部条目来搜索目录树表 DB 526, 并确定用于存储子树的 DB 服务器 6。表明所确定 DB 服务器 6 的数据被返回给相关数据创建及管理单元 510。

10 在步骤 S412, 相关数据创建及管理单元 510 向数据登记单元 502 输出 DIT 形式的相关数据以及表明所确定 DB 服务器 6 的数据。

数据登记单元 502 把输入的相关数据传送给所确定的 DB 服务器 6, 并登记数据。

15 在步骤 S416 至 S422, 从数据登记单元 502 向 UI 单元 500 通知登记结果(正常或异常结束), 并将其向用户显示。

第二 DB 程序 60

图 37 说明在图 24 所示的各个 DB 服务器 6 中运行的第二 DB 程序 60。

20 参照图 37, DB 程序 60 包括数据管理单元 600、信息 DB602、数据传送单元 604 和搜索执行单元 606。

采用这些组件, DB 程序 60 从 DB 管理服务器 5(DB 管理程序 50; 图 31)接收以 DIT 形式的目录结构表示且适当分割的相关数据的登记, 并存储该数据。

25 DB 程序 60 响应来自检索装置 40 的搜索请求而提供已登记相关数据。

DB 程序 60 在 DB 管理服务器 5 的控制下修改已登记相关数据。

在 DB 程序 60 中, 数据传送单元 604 从 DB 管理服务器 5 接收相关数据、其登记请求及其修改请求, 并将其输出到数据管理单元

600。

搜索执行单元 606 根据来自检索装置 40 的搜索条件(稍后描述且如图 38 所示的搜索程序 42 的搜索条件创建单元 420 所创建的 LDAP 操作,LDAP 操作如图 50 所示)采用数据管理单元 600 搜索相关数据。

5 此外,搜索执行单元 606 把从数据管理单元 600 得到的相关数据作为搜索结果返回给发出搜索请求的检索装置 40。

数据管理单元 600 响应来自 DB 管理服务器 5 的登记请求和修改请求而把从 DB 管理服务器 5 或者其它 DB 服务器 6 发送的相关数据(图 26 至图 28)存储在信息 DB 602 中。

10 另外,响应搜索装置 40 的检索,数据管理单元 600 从信息 DB 602 中读取相关数据,并把数据输出到搜索执行单元 606。

第一搜索程序 42

图 38 说明在图 24 所示的检索装置 40 上运行的搜索程序 42。

参照图 38,搜索程序 42 具有一种结构,其中,搜索条件创建单元 420 和搜索结果输出单元 422 被添加到图 18 所示的 DB 搜索单元 26。

采用这些组件,搜索程序 42 执行对于 DB 服务器 6 的搜索处理,并向搜索者显示作为搜索结果得到的相关数据。

20 在搜索程序 42 中,搜索操作接收器 260 接收搜索者输入到检索装置 40 的输入/输出装置 126(图 7)等中的搜索操作,并把操作输出到搜索条件创建单元 420。

搜索条件创建单元 420(目录树搜索部件)分析从搜索操作接收器 260 输入的搜索操作的内容,然后从 DB 管理服务器 5 获得与搜索操作的内容匹配的树数据(在目录树表和数据表(图 29 和图 30)中搜索所需的数据),创建与搜索操作和树数据匹配的搜索条件,以及把搜索条件输出到搜索控制单元 264。

具体来说,根据通过分析搜索操作的内容所得的相关数据的分类类型和分类值,搜索条件创建单元 420 从 DB 管理服务器 5 接收数据

表(图 30)以便搜索内容,以及获取与搜索操作的内容匹配的相关数据的顶部条目(但是,在没有这种顶部条目时,搜索条件创建单元 420 采用 DB 服务器 6-1 的顶部条目 dn_A 作为与搜索操作的内容匹配的相关数据的顶部条目)。

5 然后,搜索条件创建单元 420 搜索从 DB 管理服务器 5 接收的目录树表(图 29)中的内容,以及确定存储通过搜索数据表所得的顶部条目的 DB 服务器 6。

此外,搜索条件创建单元 420 通过使用搜索操作的内容和对从数据表所得的相关数据分类所使用的属性(分类类型和值)来创建将由
10 DB 服务器 6 执行的 LDAP 操作的命令和参数,并把命令和参数作为搜索条件输出到搜索控制单元 264。

搜索控制单元 264(数据搜索部件)把从搜索条件创建单元 420 输入的搜索条件输出到 DB 服务器 6,并控制对相关数据的搜索。

搜索结果输出单元 422 向检索装置 40 的输入/输出装置 126(图 7)
15 以及向搜索者显示/输出作为搜索结果从 DB 服务器 6 返回的相关数据。

下面将描述搜索程序 42 的整个过程。

图 39 是流程图,说明通过图 38 所示的搜索程序 42 进行的搜索过程 S44。

20 参照图 39,在步骤 S440,搜索操作接收器 260 接收来自用户的搜索条件。

在步骤 S442,搜索操作接收器 260 创建搜索请求消息(稍后参照图 50 进行描述),并把消息输出到搜索条件创建单元 420。

25 在步骤 S444,搜索条件创建单元 420 判断是否在搜索请求消息中指定分类类型和分类值。

当指定分类类型和分类值时,搜索条件创建单元 420 进入 S446 的处理,否则,进入 S454 的处理。

在步骤 S446,搜索条件创建单元 420 把分类类型和分类值发送

给 DB 管理服务器 5，并确定相应子树的顶部条目。

在步骤 S448，搜索条件创建单元 420 判断是否在 S446 的处理中已经确定子树。

5 当已经确定子树时，搜索程序 42 进入 S450 的处理，否则，进入 S454 的处理。

在步骤 S450，搜索条件创建单元 420 把所确定顶部条目发送给 DB 管理服务器 5，并确定在其中存储与顶部条目对应的子树的 DB 服务器 6。

10 在步骤 S452，搜索条件创建单元 420 把所确定的子树的顶部条目设置为搜索起始条目(查询基础)。

在步骤 S454，搜索条件创建单元 420 从 DB 管理服务器 5 接收目录树表，参考所接收的目录树表以便通过所有 DB 服务器中存储的子树把顶部条目(根条目)设置为查询基础，以及确定存储根条目的 DB 服务器 6。

15 在步骤 S456，搜索控制单元 264 从搜索条件创建单元 420 获得与搜索请求、查询基础等有关的数据，并为所确定 DB 服务器 6 创建 LDAP 操作命令。

在步骤 S458，搜索控制单元 264 通过采用已创建 LDAP 操作命令在 DB 服务器 6 上执行搜索处理，并接收和输出其结果。

20 下面将结合 DB 管理程序 50 的过程进一步描述搜索程序 42 的整个过程。

图 40 是顺序图，说明图 31 所示的 DB 管理程序 50 以及图 38 所示的搜索程序 42 的整个过程 S52。

25 参照图 40，在步骤 S520，搜索程序 42 的搜索操作接收器 260 接收用户的查询操作。

在步骤 S522，搜索操作接收器 260 把搜索请求消息传送给搜索条件创建单元 420。

在步骤 S526，搜索条件创建单元 420 把搜索请求消息中包含的

分类类型和分类值传送给 DB 管理程序 50 的相关数据创建及管理单元 510。

要注意，S526 的处理对应于在图 39 所示的步骤 S446 中从搜索条件创建单元 420 向 DB 管理服务器 5 传送分类类型/值的处理。

5 在步骤 S528，相关数据创建及管理单元 510 把从搜索条件创建单元 420 所接收的分类类型/值输出到数据表管理单元 534。

在步骤 S530，数据表管理单元 534 通过采用从相关数据创建及管理单元 510 输入的分类类型/值来参考数据表，以便获得与输入分类类型/值对应的顶部条目。

10 数据表管理单元 534 把所得顶部条目返回给相关数据创建及管理单元 510。

在步骤 S532，相关数据创建及管理单元 510 把在 S530 的处理中得到的顶部条目输出到目录树表管理单元 524。

15 在步骤 S534 和 S536，目录树表管理单元 524 通过相关数据创建及管理单元 510 把与输入顶部条目对应的 DB 服务器 6 返回给搜索条件创建单元 420。

在步骤 S538，搜索条件创建单元 420 创建搜索条件，并把搜索条件与所确定 DB 服务器 6 等的名称一起输出到搜索控制单元 264。

20 在步骤 S540，搜索控制单元 264 创建 LDAP 命令，并访问所确定 DB 服务器 6 以执行搜索。

在步骤 S542 至 S546，搜索结果从 DB 服务器 6 返回并向用户显示。

DB 系统 4 中的相关数据和搜索实例(1)

25 下面将通过举一个具体实例来描述如何登记相关数据以及如何在 DB 系统 4 中检索已登记相关数据。

图 41 是在图 24 所示的 DB 系统 4 的 DB 服务器 6 中登记的目录结构的相关数据实例的第一示意图，说明平面目录结构。

在图 41 所示的目录结构中，由图 26 所示的虚线表示的条目(ou=

关联 # 1 至 # n)不存在, 而所有相关数据直接位于根条目下面。

这个目录结构称作“平面目录结构”。

用户对于 DB 管理服务器 5(图 24)或 PC 102 的输入/输出装置 126(图 7)中显示的 GUI 图像(图 32)执行依次登记相关数据的操作。

5 通过这个操作, 下列数据以下列形式输入 DB 管理服务器 5

(关联名称, 关联类型, 名称, 角色):

(哈姆雷特的出处, 出处信息, 莎士比亚, 作者),

(哈姆雷特的出处, 出处信息, 哈姆雷特, 作品),

(威尼斯商人的出处, 出处信息, 莎士比亚, 作者),

10 (威尼斯商人的出处, 出处信息, 威尼斯商人, 作品),

(哈姆雷特的日译本, 译本信息, 哈姆雷特, 原著),

(哈姆雷特的日译本, 译本信息, 哈姆雷特的日译本, 译本),

(哈姆雷特的日译本, 译本信息, 出版、OO 出版社),

....

15 作为用户进行的登记操作的结果, 相关数据在与图 41 所示类似的 DIT 形式的目录结构中产生。

要注意, 在图 41 所示的 DIT 形式的目录结构中, 没有考虑目录信息树到子树的分割, 以及可对于整个目录信息树定义分类类型和分类值, 但平面目录结构在必要时可分割为子树。

20 在这个目录结构中, 条目设置在与五个 A 节点(“哈姆雷特的出处”, “威尼斯商人的出处”, “哈姆雷特的日译本”, “哈姆雷特的中译本”, 以及“哈姆雷特的演出”); A4, A9, A10, A19 以及 A13(参照图 19 等))对应的虚拟根条目下面。对于虚拟根条目, 分类类型和分类值被定义, 以及所定义分类类型和值在 DB 管理服务器 5
25 中根据图 30 所示的数据表来管理。

另外, 在目录结构中, 与 A 节点的关联角色对应的条目设置在与 A 节点对应的条目下面, 以及与充当相应角色的 T 节点(T41, T42, T92, T101, T103, 以及 T191; 参见图 19)有关的数据设置为条目属

性。

图 41 所示的相关数据全部存储在一个 DB 服务器 6-1(图 24)中。或者,相关数据根据顶部条目中的每个来分割,存储在五个 DB 服务器 6-1 至 6-n 中,并且成为由搜索程序 42 进行的搜索的目标。

5 图 42 和图 43 是第二和第三示意图,说明图 41 所示的目录结构的相关数据。

图 42 说明一种情况,其中,通过注意作为莎士比亚的作品的某个关联(题目),重新把相应条目“莎士比亚的作品”设置在根条目下面,以及相关数据设置在其下面,从而实现基于题目的分层结构。条目“莎士比亚的作品”对应于图 26 中虚线所示的条目(ou=关联 # 1 至 # n)之一。

图 43 说明一种情况,其中,表示分类类型“出处”、“译本”和“演出”(关联类型; A 节点的节点类型)的条目被定义并重新设置在条目“莎士比亚的作品”的下面。

15 如上所述,通过根据关联类型(节点类型)把相关数据登记在条目下面,根据关联类型(节点类型)实现分层结构。

如图 42 所示,图 41 的相关数据还可进一步分级。

也就是说,“莎士比亚的作品”可定义为图 41 所示的相关数据中包含的节点的公共题目。这样,如图 42 所示,用户可操作 DB 管理服务器 5 来创建表明条目及其下面所存储的数据与五个 A 节点对应的条目上的莎士比亚的作品相关的新条目,并且可对相关数据分级。

25 在图 42 所示的情况中,以包含条目“莎士比亚的作品”下面的条目“哈姆雷特的出处(A4)”到“哈姆雷特的演出(A13)”的目录结构来表示相关数据。

例如,条目“莎士比亚的作品”以及条目“哈姆雷特的出处(A4)”到“哈姆雷特的中译本(A19)”或其下面的条目作为子树存储在 DB 服务器 6-1 中,以及条目“哈姆雷特的演出(A13)”或其下面的条目

作为子树存储在 DB 服务器 6-2 中，等等，从而相关数据可根据目录结构被分割为子树。以这种方式得到的子树可被分割并存储在 DB 服务器 6-1 至 6-n 中。

5 在这种情况下，“莎士比亚的作品”和“哈姆雷特的演出”的目录设置为顶部条目，对于这些顶部条目定义分类类型和值，所定义分类类型和值由 DB 管理服务器 5 在图 30 所示的数据表中管理，并且其引用关系在图 29 所示的目录树表中管理。

10 在另一个实例中，在图 42 所示的相关数据的分层目录结构中，与“莎士比亚的作品”以及它下面的“哈姆雷特的出处”和“威尼斯商人的出处”对应的相关数据存储在 DB 服务器 6-1 中，与“哈姆雷特的日译本”和“哈姆雷特的中译本”对应的相关数据存储在 DB 服务器 6-2 中，以及与“哈姆雷特的演出”对应的相关数据存储在 DB 服务器 6-3 中。

15 要注意，顶部条目是子树中的根(顶部)条目，因而对于 DB 服务器 6 中存储的每个子树只定义一个顶部条目。对于这个顶部条目定义分类类型/值。

20 因此，在图 42 所示的实例中，为了在 DB 服务器 6-2 中存储与“哈姆雷特的日译本”和“哈姆雷特的中译本”对应的相关数据，提供顶部条目(例如与图 43 所示的“译本信息”对应的条目)以便结合这两个条目，并且存储其中把这个条目设置为顶部条目的子树。

以这种方式分割的子树的引用关系通过采用图 29 所示的目录表来管理，以及为各子树定义的分类类型和分类值通过采用图 30 所示的数据表来管理。

25 此外，当“出处信息”被定义为在图 41 所示的五个 A 节点之中的 A4 节点(“哈姆雷特的出处”)和 A9 节点(“威尼斯商人的出处”)的关联类型(节点类型)，“译本信息”被定义为 A10 节点(“哈姆雷特的日译本”)和 A19 节点(“威尼斯商人的中译本”)的关联类型(节点类型)，以及“演出信息”被定义为 A13 节点(“哈姆雷特的演出”)

的关联类型(节点类型)时, 所定义的关联类型(节点类型)可被分层, 如图 43 所示。

例如, 在图 43 所示的相关数据的分层结构中, 与“莎士比亚的作品”以及它下面的“出处信息”对应的相关数据存储在 DB 服务器 6-1 中, 与“译本信息”对应的相关数据存储在 DB 服务器 6-2 中, 以及与“演出信息”对应的相关数据存储在 DB 服务器 6-3 中。

在这种情况下, “莎士比亚的作品”、“译本信息”和“演出信息”的条目被设置为顶部条目, 为顶部条目定义分类类型和值, 所定义的分类类型和值在 DB 管理服务器 5 中在图 30 所示的数据表中管理, 并且其引用关系在图 29 所示的目录树表中管理。

当搜索者(用户)运行搜索程序 42(图 24)来搜索 DB 服务器 6 中存储的相关数据时, 如图 41 至 43 所示, 则搜索程序 42 分析搜索操作的内容, 以及从 DB 管理服务器 5 获得与搜索操作的内容匹配的树信息(目录树表)和数据表的必要信息(图 29 和图 30)。

此外, 搜索程序 42 创建与搜索操作和树信息匹配的搜索条件(LDAP 操作的命令和参数), 并在存储与搜索条件匹配的目录信息的 DB 服务器 6 中执行搜索, 以便获得搜索结果。

这样得到的搜索结果向搜索程序 42 的用户显示。

DB 系统 4 中的相关数据和搜索实例(2)

下面将通过表明用于商品促销的商品、商店和节目的销售信息的具体实例, 更详细地描述如何登记相关数据以及如何在 DB 系统 4 中检索已登记数据。

图 44 和图 45 是第四和第五示意图, 说明在图 24 所示的 DB 系统 4 的 DB 服务器 6 中登记的目录结构的相关数据。图 44 说明仅在特定区域、例如 Kanto 地区的销售信息, 以及图 45 说明多个区域、例如 Kanto 和 Kansai 地区的销售信息。

通过图 24 所示的 DB 管理服务器 5 或 PC 102 的用户操作, Kanto 地区的销售信息登记为图 44 所示的目录结构中表示的相关数据。

也就是说,根据用户输入的相关数据的已登记数据,结合与“销售信息”有关的目录信息的条目直接设置在虚拟根条目下面。在这个条目下面,设置与“Kanto 地区”的销售信息有关的条目。

5 对于“Kanto 地区”的销售信息,设置具有“位置信息”、“报价信息”和“参与信息”作为关联类型(节点类型)的条目。在这些条目下面,设置与诸如“商店”、“区域”、“报价”和“组织者”之类的关联角色对应的条目。作为这些条目的属性(属性存储在 DB 服务器 6 中,但没有定义或创建独立的管理表来管理属性),与充当“商店 A”、“Shinjuku”等(对应于 T 节点 ID 表(图 21)的节点 ID 和节点名称)的角色的成员有关的数据被登记。

要注意,图 44 等所示的“角色”对应于图 20 中所示的 AR 表的关联角色,以及“成员”对应于 T 节点。

在图 20 所示的实例中,对于关关节点(A 节点)A4,成员 T41 充当“作者”的角色,以及成员 T42 充当“作品”的角色。

15 如图 45 所示,当销售信息还包含除了 Kanto 地区的信息之外的信息(Kansai 地区的销售信息等)时,“销售信息”的条目直接设置在虚拟根条目下面。在这个条目下面,设置“Kanto 地区”和“Kansai 地区”的条目。在“Kansai 地区”的条目下面,设置表示“位置信息”、“报价信息”和“参与信息”的关联类型(节点类型)的条目。

20 在这些条目下面,设置与“商店”、“报价”、“区域”、“组织者”等的角色对应的条目。作为条目的属性,与充当“商店 B”、“商店 C”、“Osaka”、“活动 A”等的角色的成员有关的数据被登记。

25 例如,图 44 和图 45 所示的销售信息被分割为子树。例如,不同于虚拟根条目及其下面的“销售信息”的目录信息存储在 DB 服务器 6-1(A; 图 24)中,“Kanto 地区”及“销售信息”下级的目录信息存储在 DB 服务器 6-2(B)中,“Kanto 地区”或下级的目录信息存储在 DB 服务器 6-3(C)中,以及不同于这些地区的地区的目录(未示出)信

息存储在 DB 服务器 6-4(D)等中。

图 46 说明表明图 44 和图 45 所示的销售信息(相关数据)的顶部条目和引用条目的目录树表。

5 图 47 说明表明图 44 和图 45 所示的销售信息(相关数据)的顶部条目的分类类型和值的数据表。

DB 服务器 6-1 至 6-3 中存储的销售信息的顶部条目和引用条目、已分割目录树(或子树)的顶部条目、以及子树中存储的目录信息的分类类型/值由 DB 管理服务器 5 根据图 46 和图 47 所示的目录树表及数据表来管理。

10 图 48 说明用于图 44 和图 45 所示的销售信息的 GUI 图像。

图 48 所示的 GUI 图像在 DB 管理服务器 5 或 PC 102 的输入/输出装置 126 中显示。根据用户在 GUI 图像上的操作,在 DB 服务器 6-1 至 6-3 中登记的销售信息被搜索。

搜索仅 Kanto 地区的销售信息

15 首先描述图 44 所示的仅对 Kanto 地区的销售信息的搜索。

但是要注意,图 44 说明仅 Kanto 地区的销售信息,但作为实际的目录信息,其它地区的信息以及除销售信息以外的信息都被登记/存储。

20 搜索者(用户)在检索装置 40 的输入/输出装置 126 所显示的 GUI 图像(图 48)上执行搜索例如“通过参与 Kanto 地区的某个销售节目而与商店 B 联系的商店”的操作。

图 49 说明一个过程,其中,图 24 所示的检索装置 40 从图 46 和图 47 所示的目录树表及数据表中检索目标 DB 服务器 6。

25 例如,如图 49 所示,检索装置 40 分析搜索者的搜索请求的内容,并参考从 DB 管理服务器 5 得到的树信息中包含的数据表(图 47)的内容,以便得到其中“地区”和“Kanto 地区”作为分类类型和分类值被相关的子树(“Kanto 地区”的目录)的顶部条目 dn_B。

此外,检索装置 40 还搜索树信息中包含的目录树表(图 46)的内

容, 以及检索顶部条目 dn_B 已经存储在其中的 DB 服务器 6-2(B)。

图 50 说明创建从图 48 所示的搜索条件中得到的并用于搜索图 44 所示的仅 Kanto 地区的销售信息的 LDAP 操作的搜索请求消息。

5 要注意, 图 50 说明用户以 XML 消息输入的搜索请求, 并且用于实际搜索的 LDAP 操作命令从这个搜索请求来产生, 但 XML 以外的格式并不排除在本发明的范围之外。

用于搜索“通过参与 Kanto 地区的某个销售节目与商店 B 联系的商店”的 LDAP 操作(LDAP 命令和参数)从操作员所输入的搜索条件、从树信息得到的子树的顶部条目以及表明它存储在 DB 服务器 6-2(B)的信息中产生, 以及搜索在 DB 服务器 6-2(B)中执行。

图 50 说明用于创建 LDAP 操作的搜索请求消息, 它表明检索装置 40 在存储 Kanto 地区的目录以及此目录下面的条目的 DB 服务器 6-2(B; 图 24)中执行以下搜索操作(1)和(2)。

15 (1) 其下级条目“角色”为“参加者”以及其下级条目“成员”为“商店 B”的条目从其中“关联类型”(图 22 所示的 A 节点的节点类型; 关联类型)为“参与信息”的所有条目中获得, 以及成员属性值(成员名称)从其中“角色”为“组织者”的下级条目的那些中获得。

作为这个处理的结果, 得到“节目 A”, 作为商店 B 参与的销售节目。

20 (2) 其下级条目“角色”为“组织者”以及其下级条目“成员”为“节目 A”的条目从其中“关联类型”为“参与信息”的所有条目中获得, 以及成员属性值(成员名称)从其中“角色”为“参加者”的下级条目的那些中获得。

25 作为这个处理的结果, 得到“商店 A”和“商店 B”, 作为参与销售节目“节目 A”的商店。检索装置 40 选择不同于搜索操作(图 48)的内容中包含的“商店 B”的“商店 A”, 作为最终搜索结果。

搜索特定区域中的销售信息

随后将描述搜索图 45 所示的 Kanto 及其它地区的销售信息。

搜索者(用户)在检索装置 40 的输入/输出装置 126 中显示的 GUI 图像(图 48)上执行搜索例如“不管地区(在所有地区)通过对商品报价来参与商店 A 参加的活动的商店”的操作。

5 检索装置 40 分析搜索者的搜索操作的内容,并搜索从 DB 管理服务器 5 得到的树信息中包含的数据表(图 47)。但是,由于搜索操作的内容不限地区,因此检索装置 40 检索由“Kanto 地区”、“Kansai 地区”等的地区条目所引用的“销售信息”的条目(例如在图 46 中为 dn_A)。

10 此外,检索装置 40 还搜索目录树表(图 46),以及检索顶部条目 dn_A 已经存储在其中的 DB 服务器 6-1(A)。

搜索“不管地区(在所有地区)通过对商品报价来参与商店 A 参加的活动的商店”的 LDAP 操作(LDAP 命令和参数)从操作员所输入的搜索条件、从树信息得到的顶部条目以及表明它存储在 DB 服务器 6-1(A)的信息中产生,以及搜索在 DB 服务器 6 中执行。

15 LDAP 操作包括以下搜索操作。

(1) 其下级条目“成员”为“商店 A”以及其下级条目“角色”为“被报价方”的条目从其中“关联类型(图 22 所示的 A 节点的节点类型; 关联类型)”为“报价信息”的所有条目中获得,以及成员属性值(成员名称)从其中“角色”为“报价”的下级条目中获得。

20 作为这个处理的结果,得到“商品 A”,作为由商店 A 报价的商品。

(2) 其下级条目“成员”为“商品 A”以及其下级条目“角色”为“参加者”的条目从其中“关联类型”为“参与信息”的所有条目中获得,以及成员属性值(成员名称)从其中“角色”为“组织者”的下级条目中获得。

25 作为这个处理的结果,得到“活动 A”,作为商品 A 被报价的活动节目。

(3) 其下级条目“成员”为“活动 A”以及“角色”为“组织者”

的条目从其中“关联类型”为“参与信息”的所有条目中获得，以及成员属性值(成员名称)从其中“角色”为“参加者”的下级条目的那些中获得。

5 作为这个处理的结果，检索装置 40 检索“商品 A”和“商品 C”，作为由参与活动 A 的商店报价的商品。

(4) 其下级条目“成员”为“商品 A”以及其下级条目“角色”为“报价”或者其下级条目“成员”为“商品 C”且其下级条目“角色”为“报价”的条目从其中“关联类型”为“报价信息”的所有条目中获得，以及成员属性值(成员名称)从其中“角色”为“被报价方”
10 的下级条目的那些中获得。

作为这个处理的结果，得到“商店 A”和“商店 C”，作为向活动 A 对商品 A 和 C 报价的商店，以及检索装置 40 从“商店 A”和“商店 C”中选择不同于搜索操作(图 48)的内容中包含的“商店 A”的“商店 C”，作为最终搜索结果。

15 目录结构以及 DB 服务器之间的负荷分布的动态修改

如上所述，在 DB 系统 4(图 24)中，相关数据以目录结构来表示。因此，易于执行在 DB 服务器 6 中基于子树的分割和复制。

因此，在 DB 系统 4 中，例如，即使在对 DB 服务器 6 中存储的某个条目的访问因某种原因变得过多时，易于对负荷分布采取若干措施，例如添加用于存储条目的复制服务器(镜像服务器)，以及把条目
20 及其下级条目(某个条目及其下级条目的集合以下可称作子树)共同传送到另外的 DB 服务器 6。

下面将对通过采用 DB 系统 4 的特征的目录结构的动态修改和适合于 DB 服务器 6 之间的负荷分布以及防止 DB 系统 4 的系统性能下降的子树交换/移动进行描述。
25

下列方法用于对目录结构的动态修改以及在 DB 系统 4 中的 DB 服务器 6 之间的子树交换/移动：

(1) 在相同 DB 服务器 6 中的新子树的创建(新子树的提供)，以

及

(2) 子树从高负荷 DB 服务器 6 到低负荷 DB 服务器 6 的传送，或者高负荷 DB 服务器 6 的频繁访问的子树与低负荷 DB 服务器 6 的不频繁访问的子树之间的交换(子树交换/移动)。

5 新子树的提供

首先将描述 DB 系统 4 中的新子树的提供。

图 51 说明在图 24 所示的 DB 系统 4 的 DB 服务器 6 中的新子树的提供，其中，(A)部分说明提供新子树之前的目录结构，以及(B)部分说明提供新子树之后的目录结构。

10 将考虑如图 51(A)所示的一种情况，其中，当检索装置 40 把 queryBaseDN 设置为查询基础并通过采用过滤器 queryFiler 来执行搜索时，与搜索条件匹配的条目集中于某个 DB 服务器 6 所存储的目录树(或子树)的一部分中。

15 要注意，过滤器 queryFiler 对应于 LDAP 的搜索条件，以及条件“角色” = “报价”、“成员” = “商品 C”等构成到目前为止所述的搜索操作中的过滤器 queryFiler。

在过滤器 queryFiler 中，复杂条件可通过一些条件的逻辑积来表示。

20 DB 服务器 6 计算对于每个搜索操作，各条目与搜索条件匹配的次数，以及这种计数的结果在 DB 管理服务器 5 中合计，从而在特定搜索条件下可得到一组频繁访问的条目及其上级条目。

例如，在提供新子树之前的目录结构(图 51(A))中，通过执行这种测量，能够检测到附图所示的加阴影条目匹配搜索条件,以及在某些搜索条件(查询基础: queryBaseDN, 过滤器: queryFilter)下访问集中
25 在这些条目上。

当检测到这种访问集中时，把上级条目(groupDN)提供给一组访问集中的条目，以及那些条目将被构建为新子树。这样，对该条目集合的访问可被局部化，并且可防止 DB 系统的性能的下降。

此外,还可准备在一个 DB 服务器 6 与另一个之间移动访问集中的子树。

图 52 说明用于图 51(A)和(B)所示的新子树的提供的搜索条件管理表。

5 当通过提供上级条目(groupDN)新建子树时,DB 管理服务器 5(图 24)创建其中把搜索起始条目(查询基础; queryBaseDN)、用于搜索的过滤器(queryFilter)以及在搜索起始条目之下并建立在访问集中条目之上的条目(新查询基础; groupDN)相关的搜索条件管理表,如图 52 所示,并把该表提供给检索装置 40。

10 在接收搜索条件管理表以便创建搜索条件的情况下,当在其查询过滤器中包括上述特定过滤器(queryFilter)作为逻辑积的搜索在已改变条目(查询基础: queryBaseDN)之下的子树中进行时,检索装置 40 通过采用排除了不是用于原始搜索起始条目(查询基础; queryBaseDN)之下的子树、而是用于新建搜索起始条目(新查询基础; groupDN)之下的子树的特定过滤器(queryFilter)的过滤器来执行搜索。因此能够缩短搜索所需的时间。

15 当过滤器不包含特定过滤器(queryFilter)时,原始条目(queryBaseDN)被设置为搜索起始条目,以及搜索在所有下级子树的范围中被执行。

20 在提供新子树之后,如没有使用搜索条件管理表的情况那样,当搜索的顶部条目无法从数据表(图 47)或者搜索条件管理表中得到时,检索装置 40 在 DB 服务器 6-1 至 6-n(A 至 N)中从顶部目录执行搜索。

子树的移动/交换

25 图 53(A)和图 53(B)举例说明图 24 所示的 DB 系统 4 的 DB 服务器 6-1 与 6-2(A 与 B)之间的子树交换/移动,其中图 53(A)说明子树交换之前的目录结构,以及图 53(B)部分说明子树交换之后的目录结构。

将考虑如图 53(A)所示的一种情况, 其中存在访问集中于 DB 服务器 6-2(B)的子树(条目 dn_3 或下级的子树), 同时存在 DB 服务器 6-1(A)中几乎没有访问的子树(条目 dn_2 或下级的子树)。

5 对每个条目的访问次数由 DB 服务器 6 计算, 以及这种计数的结果由 DB 管理服务器 5 合计, 从而可得到频繁访问的子树。同时可得到 DB 服务器 6 的负荷状态。另外, 访问集中的子树可通过采用图 52 所示的搜索条件管理表来获得。

10 当检测到这种访问集中(不均匀性)时, 如图 53(B)所示, 高负荷 DB 服务器 6-2(B)的条目 dn_3 或下级的子树配置为 DB 服务器 6-1(A)的条目 dn_A-1 或下级的子树, 而 DB 服务器 6-1(A)的条目 dn_2 的子树则配置为其中 DB 服务器 6-2(B)的 dn_2 为顶部条目的子树, 以及子树在 DB 服务器 6-1 与 6-2(A 与 B)之间交换。因此, 负荷可在 DB 服务器 6-1 与 6-2(A 与 B)之间分布。

15 但是, 宏观地考虑访问集中/不均匀性(例如, 被搜索的次数少、从而降低访问和负荷的情况), 并不是始终需要将搜索条件管理表用于子树移动/交换。

新子树(搜索条件管理表)的提供被设计以根据在某个时间点之前出现的在特定搜索条件下访问集中的程度来实现之后的高速(高效)搜索处理。

20 另一方面, 子树移动/交换解决了以下问题: 负荷的极不均匀分布因不取决于搜索条件或者不限于特定搜索条件的对于各服务器或子树的访问的绝对次数(频率)的增加/减少而出现。

25 图 54 说明由图 53(A)和(B)所示的子树交换所引起的对 DB 服务器 6-1 和 6-2(A 和 B)的目录树表(图 46)的更新, 其中, (A)部分说明更新前的目录表, 以及(B)部分说明更新后的目录表。

要注意, 这种子树交换对于如图 51(A)和(B)所示的新增子树以及现有子树来执行。

当执行更新时, DB 管理服务器 5 更新 DB 服务器 6-1 和 6-2(A

和 B)的目录树表的内容,如图 54(A)和(B)所示(但是,在本例中,DB 服务器 6-1(A)的目录树表的内容没有变化)。

例如,当访问集中于图 53(A)所示的 DB 服务器 6-1(A)的目录 dn_1,而对整个 DB 服务器 6-2(B)的访问不频繁从而设置低负荷状态时,仅通过把目录 dn_1 或下级的子树从 DB 服务器 6-1(A)移动到 DB 服务器 6-2(B),在 DB 服务器 6 之间分布负荷。

上述子树移动/交换根据以下过程来执行。

(1) 访问频率的测量: DB 服务器 6 测量对于从紧接虚拟根条目之下的条目到比相关数据的第一级基本成分(图 26)高一层位置中的条目的条目的访问次数(频率),以及 DB 管理服务器 5 合计测量值。

(2) 移动/交换源的子树的选择:在各 DB 服务器 6 中检测到对属于相同级的多个条目之下的子树的极不均匀访问(集中)时,DB 管理服务器 5 选择这些条目之下的子树作为移动/交换源的子树。

(3) 移动/交换目标的 DB 服务器的选择:在选择移动/交换源的子树时,DB 管理服务器 5 选择其中没有作为下级条目来存储移动/交换源的子树并且其访问频率(负荷)整体上低于已经存储了移动/交换源的子树的 DB 服务器 6 的其它 DB 服务器 6。

第三 DB 程序 62

图 55 说明在图 24 所示的 DB 系统 4 中新建或移动/交换子树时在 DB 服务器 6 中运行的第三 DB 程序 62。

参照图 55,DB 程序 62 采用一种配置:数据传送单元 620 和访问监测单元 624 被添加到图 37 所示的第二 DB 程序 60 中。

采用这些组件,除了与第二 DB 程序 60 相似的那些功能之外,DB 程序 62 还实现提供新子树和移动/交换所需的功能。

在 DB 程序 62 中,数据传送单元 620 根据来自 DB 管理服务器 5 的控制来传送图 53(A)和 53(B)所示的子树移动/交换所需的相关数据。

访问监测单元 624 测量对于 DB 服务器 6 中存储的目录信息的各

条目的访问的次数(频率), 并把测量值传送给 DB 管理服务器 5。

第二 DB 管理程序 56

图 56 说明在图 24 所示的 DB 系统 4 中新建和移动/交换子树时在 DB 管理服务器 5 中运行的第二 DB 管理程序 56。

5 参照图 56, 第二 DB 程序 56 采用一种体系结构: 重新配置过程单元 560、数据传输控制单元 562(数据传输部件)、DB 监测单元 570、监测 DB 572、搜索条件管理单元 580、搜索条件 DB 582 以及搜索条件提供单元 584 被添加到图 31 所示的第一 DB 管理程序 50 中。

10 采用这些组件, 除了与第一 DB 管理服务器 50 相似的那些功能之外, DB 管理程序 56 还实现提供新子树和移动/交换所需的功能。

 在 DB 管理程序 56 中, DB 监测单元 570 合计从 DB 服务器 6 发送的对条目的访问次数(频率)以及与用于访问的 LDAP 的搜索操作有关的数据(查询基础以及过滤器), 并把合计值存储在监测 DB 572 中。

15 DB 监测单元 570 把所存储的访问频率合计值以及与搜索操作有关的数据输出到重新配置过程单元 560。

 重新配置过程单元 560 根据从 DB 监测单元 570 输入的对条目的访问次数的合计值来控制数据传输控制单元 562, 并执行如图 51(A)和(B)所示的提供新子树的处理。

20 重新配置过程单元 560 根据对条目的访问次数的合计值来控制数据传输控制单元 562, 并执行如图 53(A)和(B)所示的子树的移动/交换所需的处理。

 此外, 重新配置过程单元 560 还控制数据表管理单元 534 来执行如图 54(A)和(B)所示的提供新子树以及移动/交换所引起的目录树表(图 46)的修改所需的处理。

25 搜索条件管理单元 580 在新建或移动/改变子树时创建图 52 所示的搜索条件管理表, 并把该表存储在搜索条件 DB 582 中。

 另外, 搜索条件管理单元 580 还把所存储搜索条件管理表输出到搜索条件提供单元 584。

搜索条件提供单元 584 把从搜索条件管理单元 580 输入的搜索条件管理表提供给检索装置 40(搜索程序 44; 图 59)。

下面将更详细地描述访问次数的计数。

图 57 是流程图, 说明图 56 所示的 DB 管理程序 56 的过程 S56。

5 参照图 57, 在步骤 S560, DB 管理程序 50 执行初始化。

在步骤 S562, DB 监测单元 570 开始测量对 DB 服务器 6 的各条目的访问次数。

在步骤 S564, DB 服务器 6 从搜索控制单元 264 接收 LDAP 搜索操作命令。

10 在步骤 S566, DB 监测单元 570 检索输入到 DB 服务器 6 的搜索操作命令中包含的搜索条件。

在步骤 S568, DB 监测单元 570 从 DB 服务器 6 中检索与搜索条件匹配的条目的数据。

15 要注意, 执行 S564 至 S568 的处理以便合计与 DB 服务器 6 所执行的搜索操作有关的数据, 以及合计值从 DB 服务器 6 发送给 DB 监测单元 570。

在步骤 S570, DB 监测单元 570 判断是否经过了合计与搜索操作有关的数据的预定测量时间。

20 DB 管理程序 56 在经过了测量时间时进入 S572 的处理, 否则进入 S564 的处理。

在步骤 S572, DB 监测单元 570 结束访问次数的测量。

在步骤 S574, DB 监测单元 570 在测量时间内从测量的开始到结束计算对所有条目的访问次数以及所有 DB 服务器 6 的负荷状态。

25 在步骤 S576, 重新配置过程单元 560 根据 DB 监测单元 570 所计算的测量结果来检测是否存在对所有条目的访问集中。

在步骤 S578, DB 监测单元 570 判断是否继续此过程。

如果继续此过程, DB 管理程序 56 进入 S560 的处理, 否则结束此过程。

图 58 说明在多个 DB 服务器上存储的相同层的条目。

例如, 某个条目或子树上的访问集中通过下列方法(1)或(2)来确定。

5 (1) 在预定测量时间内对所有搜索操作计算访问集中指数(访问集中指数 = 满足搜索条件的条目的数量/特定层存在的条目数量(图 51(A)))。

当每个搜索操作的执行次数大于预设数值, 并且所计算的访问集中指数等于或小于门限值时, 确定出现相对于搜索条件的访问集中。

10 (2) 存储分割子树的 DB 服务器 6 被设置为测量的目标, 在其中各子树的顶部条目以及属于与顶部条目相同层的所有条目用作根据搜索请求的查询基础的处理的次数(频率)被设置为对子树的访问频率, 以及访问集中由类似于(1)的方法来确定。

第二搜索程序 44

15 图 59 说明在图 24 所示的 DB 系统 4 中新提供和移动/交换子树时在检索装置 40 上运行的第二搜索程序 44。

参照图 59, 第二搜索程序 44 采用一种配置: 搜索条件变换单元 440 被添加到图 38 所示的第一检索装置 40 中。

除了与第一检索装置 40 相似的那些功能之外, 搜索程序 44 还通过采用搜索条件管理表来变换搜索条件。

20 在搜索程序 44 中, 搜索条件变换单元 440 从 DB 管理单元 5 接收搜索条件管理表(图 52), 并判断搜索条件创建单元 420 创建的搜索条件中包含的过滤器和搜索起始条目(查询基础)是否匹配搜索条件管理表中表示的条件。

25 当搜索条件中包含的过滤器和查询基础匹配搜索条件管理表中表示的条件时, 搜索条件变换单元 440 把原始搜索起始条目(查询基础; queryBaseDN)修改为新的搜索起始条目(新查询基础; groupDN), 并且控制搜索条件创建单元 420 把过滤器变换为新的过滤器, 其中, 搜索条件管理表中表示的过滤器(queryFilter)从搜索条件的过滤器中

删除。

新建子树时 DB 系统 4 的操作

下面将描述新建子树时 DB 系统 4 的整个操作。

5 如以上参照图 51(A)所述, 对于每个搜索操作、条目匹配搜索条件的次数由 DB 服务器 6 计数, 以及这类计数结果由 DB 管理服务器 5 合计, 从而可得到在特定搜索条件下的一组频繁访问的条目及其顶部条目。

10 当这种访问集中被确定时, 通过在访问集中的条目集合中提供顶部条目(groupDN), 以及把其下级条目构造为新子树, 就能够使对条目集合的访问局部化, 并且防止 DB 系统的性能的下降。

DB 服务器 6 通知 DB 管理服务器 5 关于对各条目的访问次数以及用于访问的过滤器的情況。

DB 管理服务器 5 从 DB 服务器 6 计算对条目访问的次数(频率)以及用于访问的过滤器。

15 在对于某个 DB 服务器 6 的子树的访问频繁时, 如图 51(A)和(B)所示, DB 管理服务器 5 控制 DB 服务器 6 来创建新的子树。

DB 管理服务器 5 把搜索条件管理表(图 52)的内容修改为匹配新建子树。

20 DB 管理服务器 5 把已修改的搜索条件管理表提供给检索装置 40。

在响应搜索者(用户)的搜索请求而根据搜索操作创建搜索条件时, 检索装置 40 通过采用从 DB 管理服务器 5 提供的搜索条件管理表来执行变换, 并且执行 DB 服务器 6 中的搜索。

移动/交换子树时 DB 系统 4 的操作

25 下面将描述移动/交换子树时 DB 系统 4 的整个操作。

如以上参照图 58 所述, 通过合计各条目被设置为多个 DB 服务器上的相同层的条目的查询基础的次数, 可确定子树上的访问集中。

DB 服务器 6 测量每个条目用作查询基础的次数, 并将结果通知

DB 管理服务器 5。

DB 管理服务器 5 计算各条目被设置为各 DB 服务器 6 的顶部条目以及与顶部条目相同层的其它 DB 服务器的所有条目的查询基础的次数，并计算每个子树的访问频率。

5 在对于某个 DB 服务器 6 的子树的访问频繁时，如图 53(A)和(B)所示，DB 管理服务器 5 控制 DB 服务器 6 来移动/交换子树。

另外，DB 管理服务器 5 把目录树表(图 46)、数据表(图 47)等的内容修改为匹配所移动/交换的子树。

DB 管理服务器 5 把已修改目录树和数据表提供给检索装置 40。

10 检索装置 40 通过采用所提供的目录树和数据表，根据搜索者(用户)的搜索操作来执行 DB 服务器 6 中的搜索。

应当注意，虽然本发明的一些公开实施例以包含在计算机可读介质中并且可由计算机执行的软件指令的形式来实现，但本发明不限于这种配置。在备选实施例中，硬连线电路可以用来代替软件指令或者
15 与其结合。因此，本发明不限于硬件电路和软件的任何特定组合。

还应当注意，本发明的公开实施例提供以下优点。

(1) 公开实施例提供允许轻易添加各种信息、特别是数据元素或字段而无需改变数据库模式的数据模型及数据库系统。

(2) 公开实施例还提供允许数据被唯一分析、存储以及检索供再
20 生的数据模型和数据库系统。

(3) 公开实施例还提供允许负荷容易地在多个处理装置之间分布的数据库系统和管理方法及软件。

(4) 公开实施例还提供允许用户搜索已登记信息的数据模型和数据库系统以及管理方法及软件。

25 (5) 公开实施例还提供把预先存在的数据库转换为允许轻易添加各种信息、特别是数据元素或字段而无需改变数据库模式的数据模型及数据库系统的方法。

虽然已经描述和说明了本发明的具体实施例，但大家很清楚，可

进行对具体说明和描述的实施例的细节上的变更,而没有明确背离所附权利要求中定义的本发明的真实精神和范围。

图 1

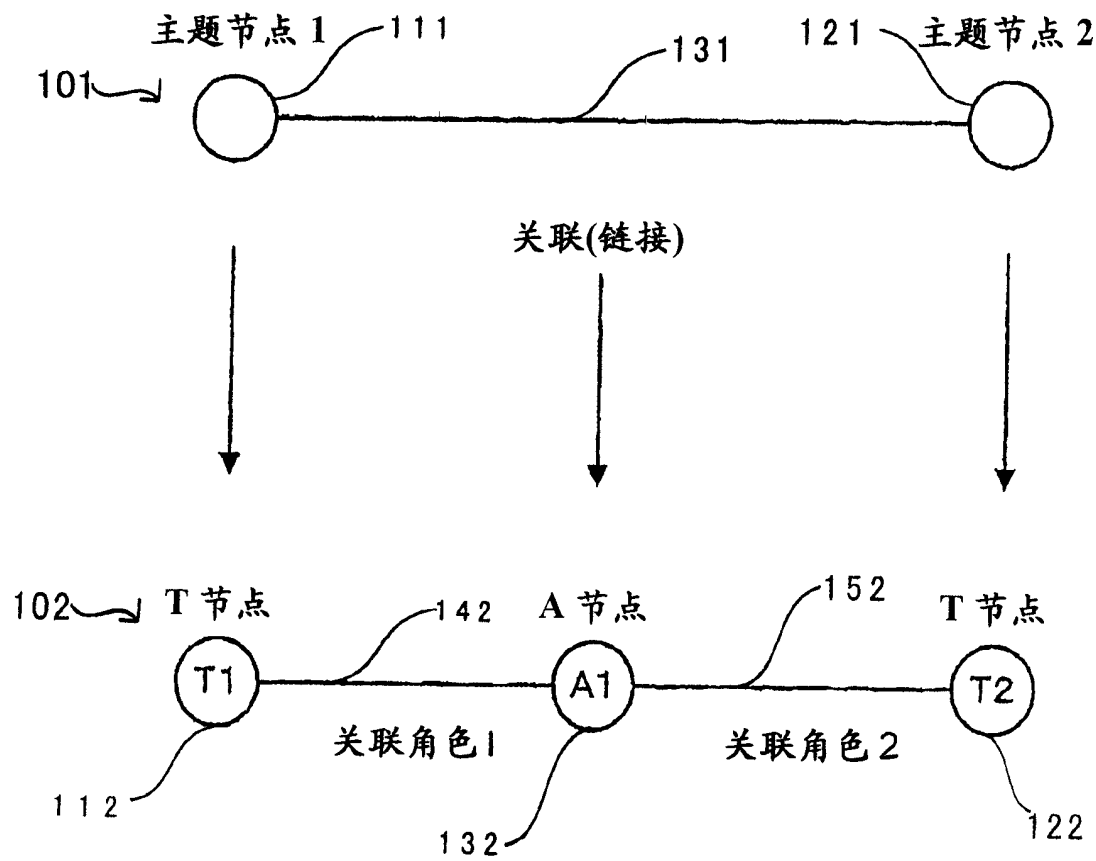


图 2

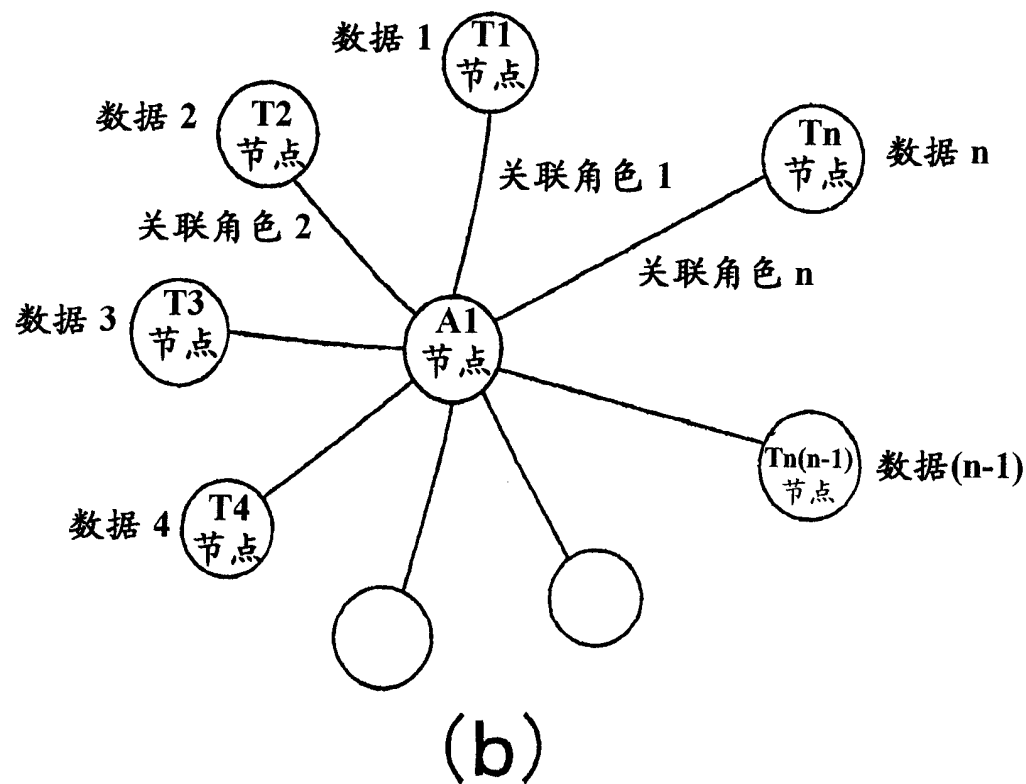
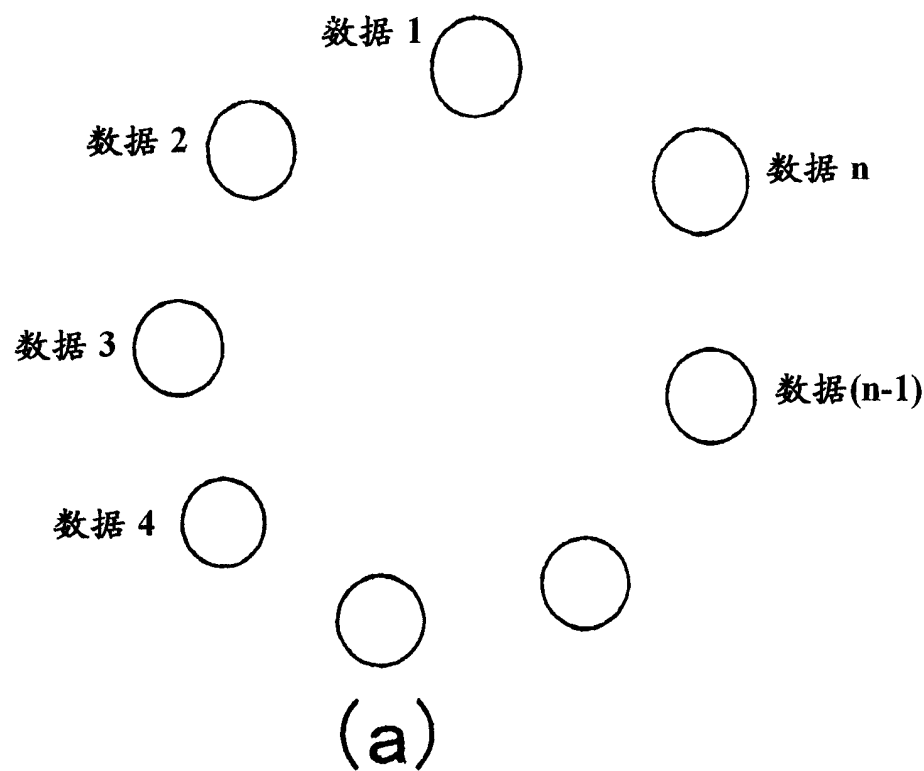
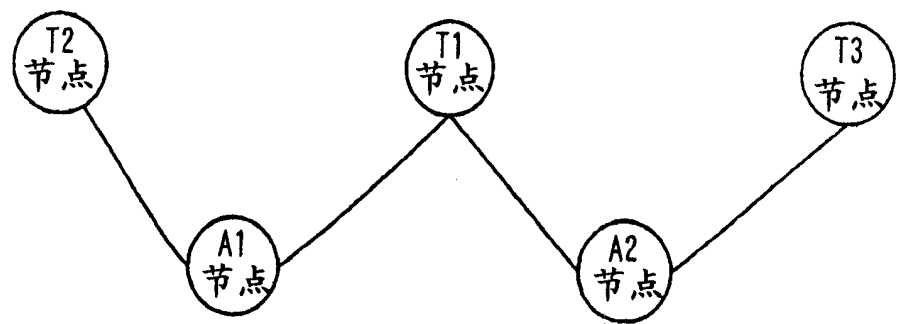
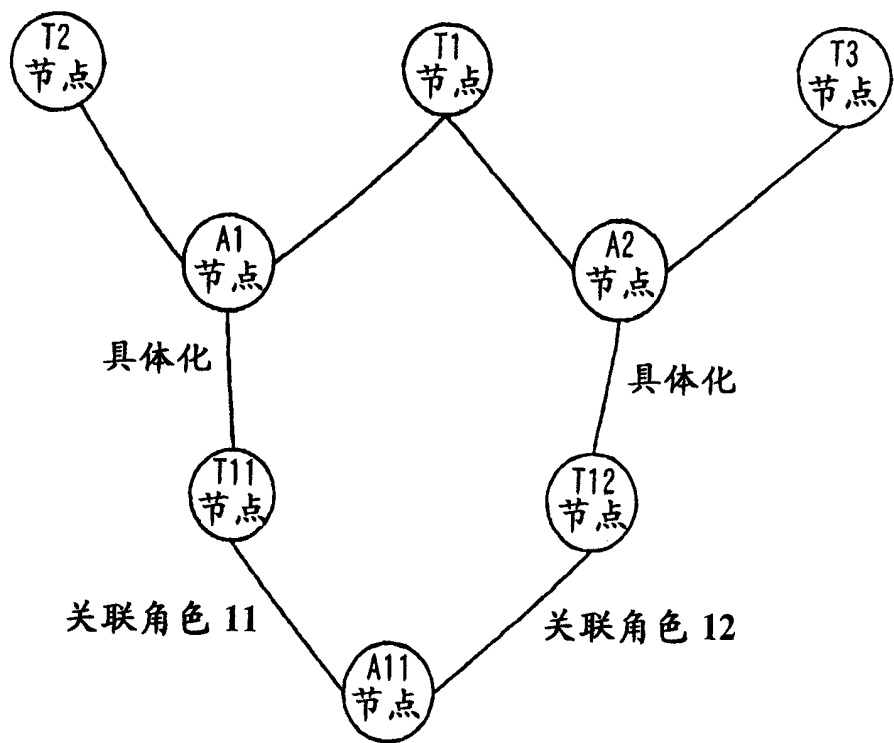


图 3



(a)



(b)

图 4

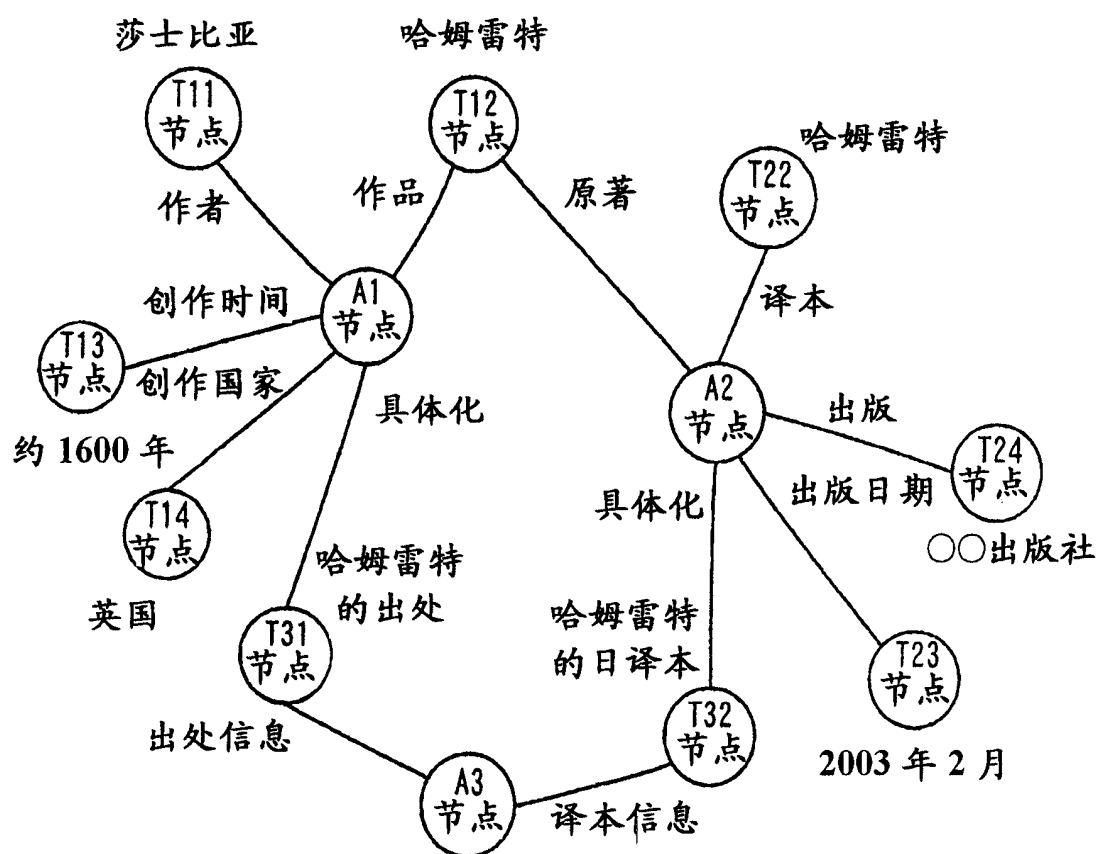


图 5

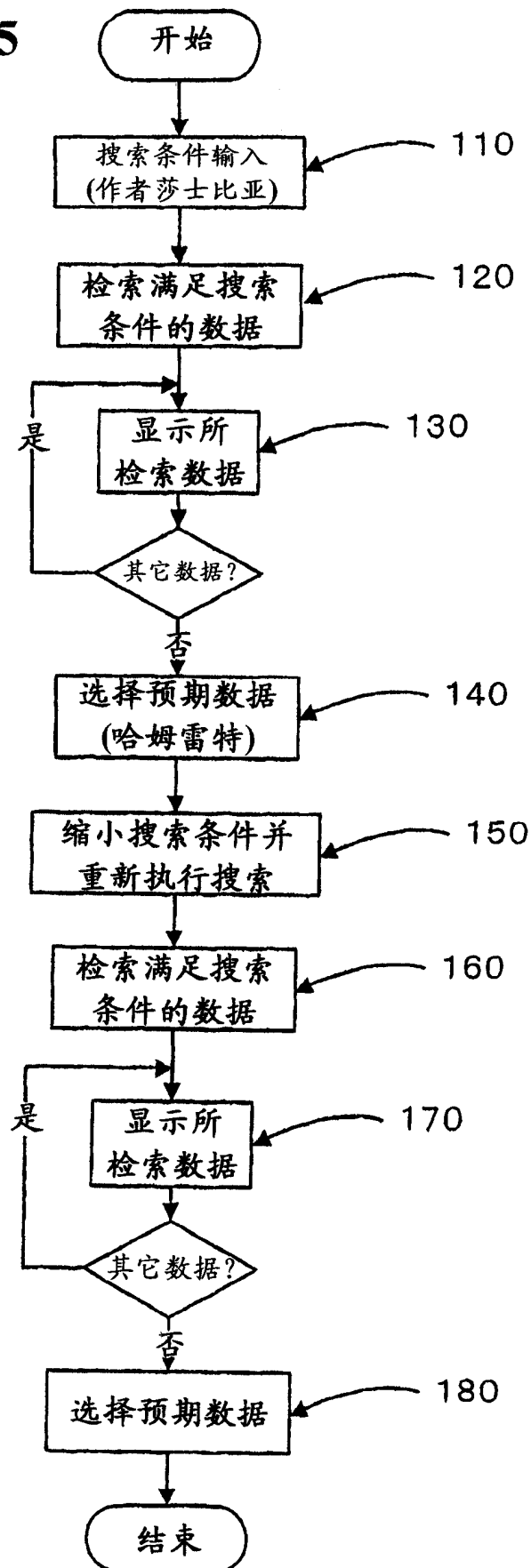
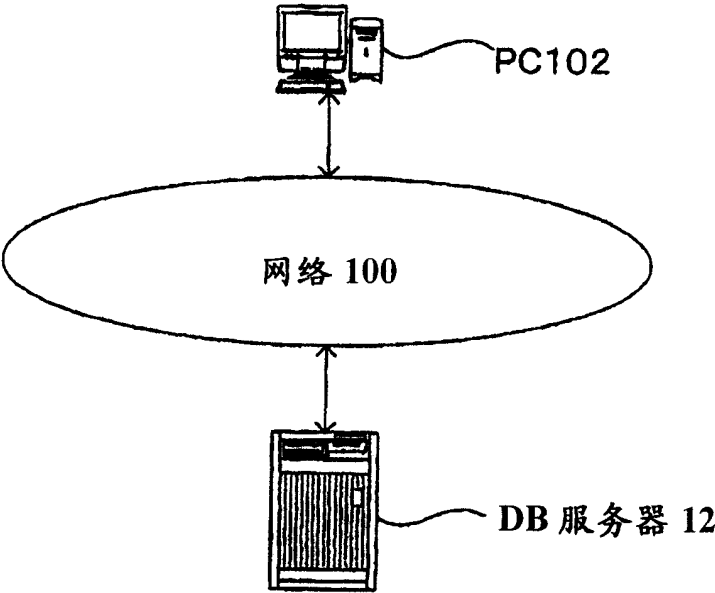


图 6



DB 系统 1

图 7

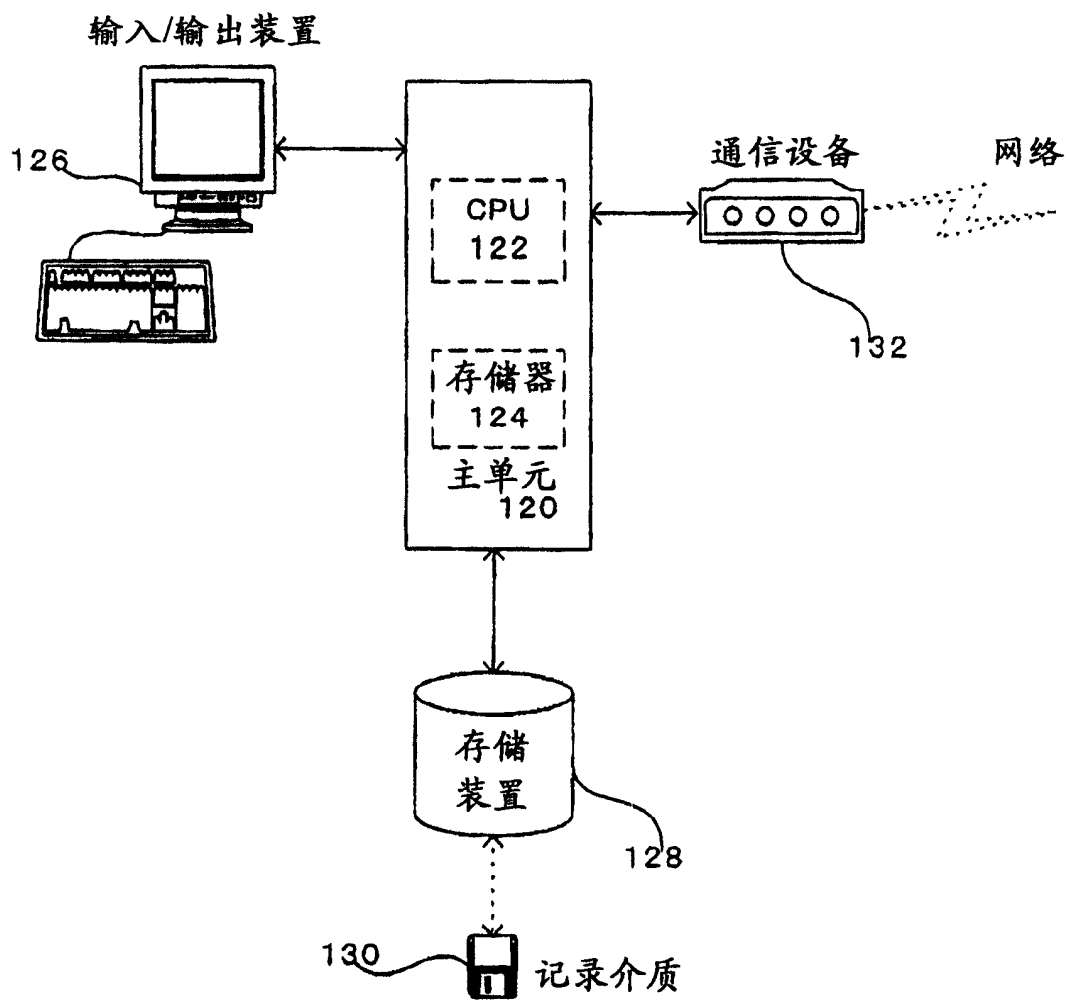
12. 102(30. 32. 40. 5. 6)

图 8

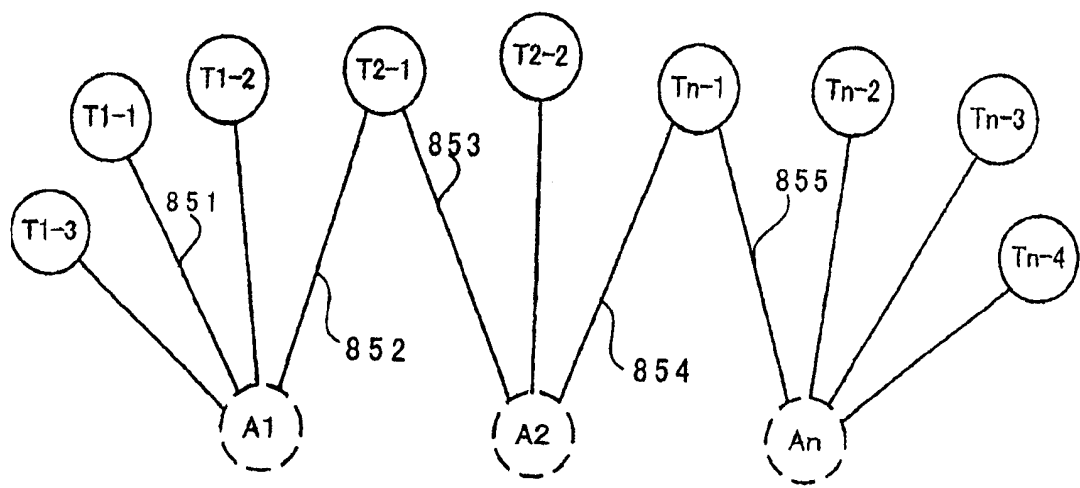
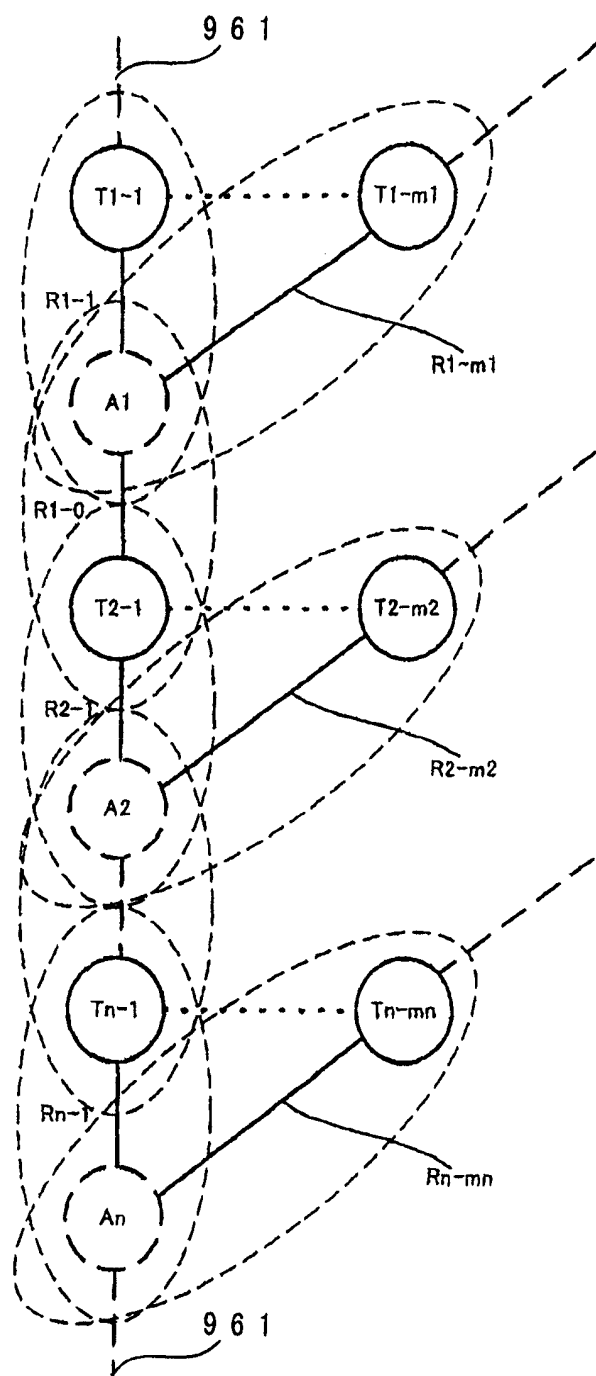


图 9



-  AR表中存储的数据
 (A节点ID, T节点ID, 关联角色(R))
-   ID表中存储的数据
 (T节点ID, 属性(节点类型(NT)), 名称(节点名称(N)))
 (A节点ID, 属性(节点类型(NT')), 名称(节点名称(N'))))

图 10

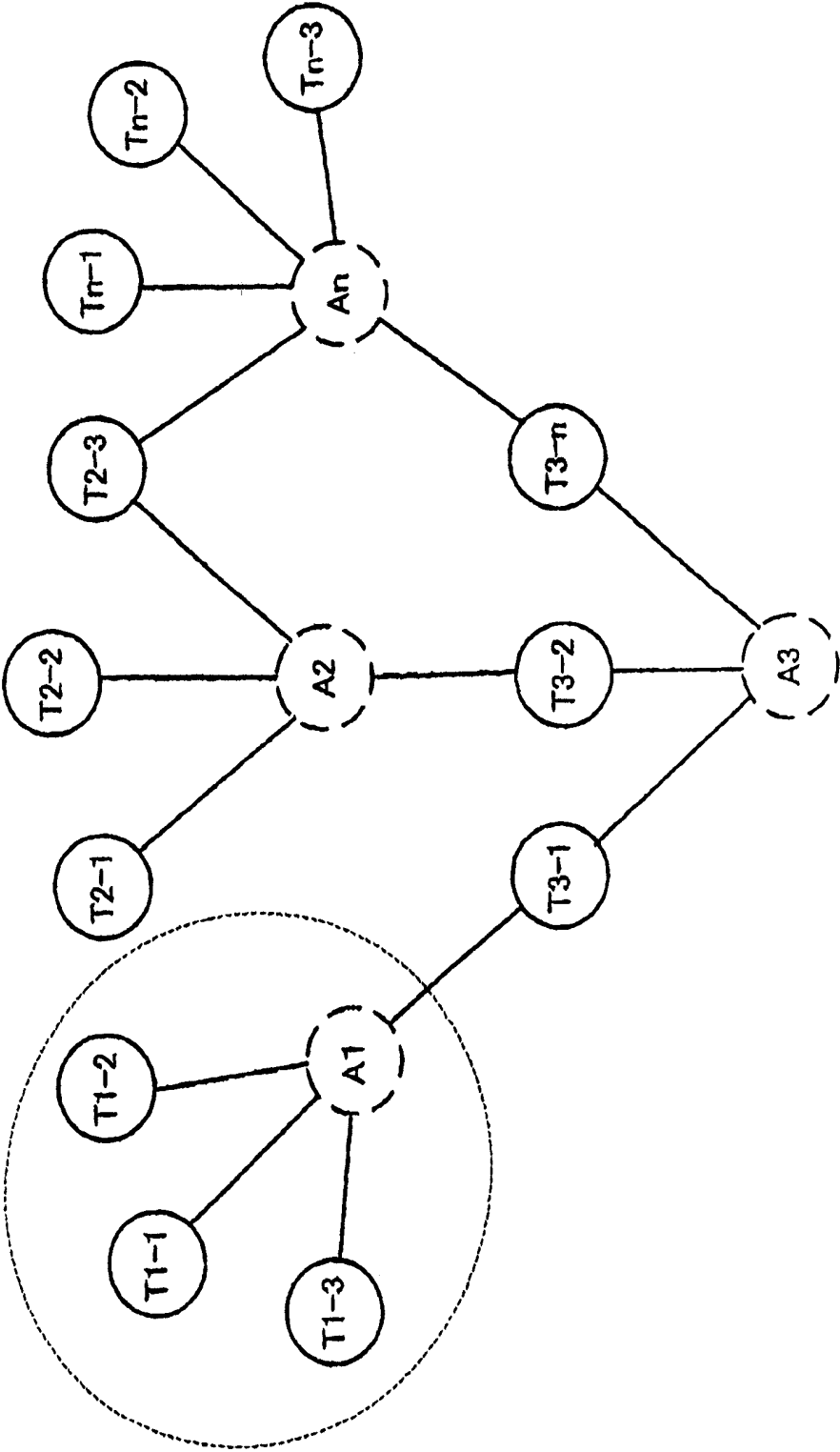


图 11

AR 表

A 节点 ID	T 节点 ID (节点名称(N))	关联角色 R
A1	T1-1	R1-1
A1	T1-m1	R1-m1
A1	T2-1	R1-0
A2	T2-1	R2-1
A2	T2-m2	R2-m2
An	Tn-1	Rn-1
An	Tn-mn	Rn-mn

图 12

ID 表(用于 T 节点)

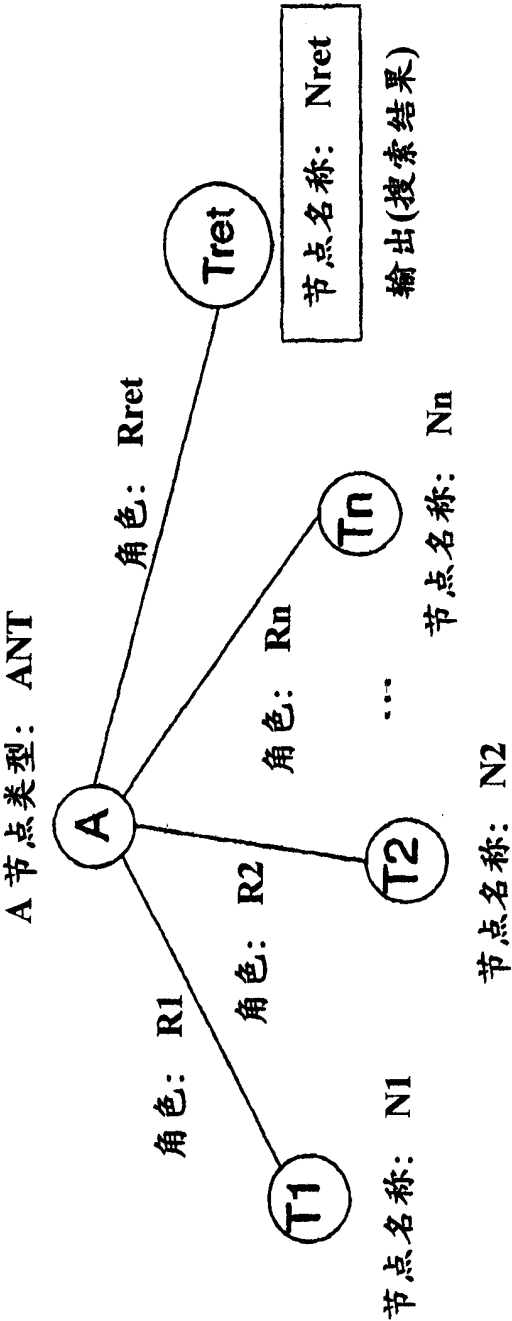
T 节点 ID	节点类型(NT)	节点名称(N)
T1-1	NT1-1	N1-1
T1-m1	NT1-m1	N1-m1
T2-1	NT2-1	N2-1
T2-m2	NT2-m2	N2-m2
Tn-1	NTn-1	Nn-1
Tn-mn	NTn-mn	Nn-mn

图 13

ID 表(用于 A 节点)

A 节点 ID	节点类型(NT)	节点名称(N)
A1	NT1	N1
A2	NT2	N2
An	NTn	Nn

图 14



搜索条件 (Rret, (ANT1, ANT2, ...), Filter) → 搜索结果 (Tret, Nret)

$$\text{Filter} = ((R1, N1), (R2, N2), ..., (Rn, Nn))$$

图 15

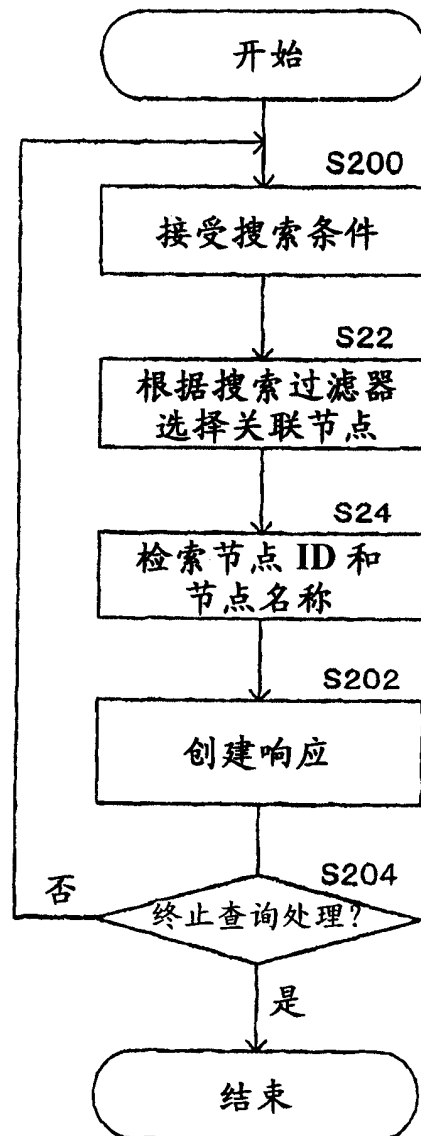
S20

图 16

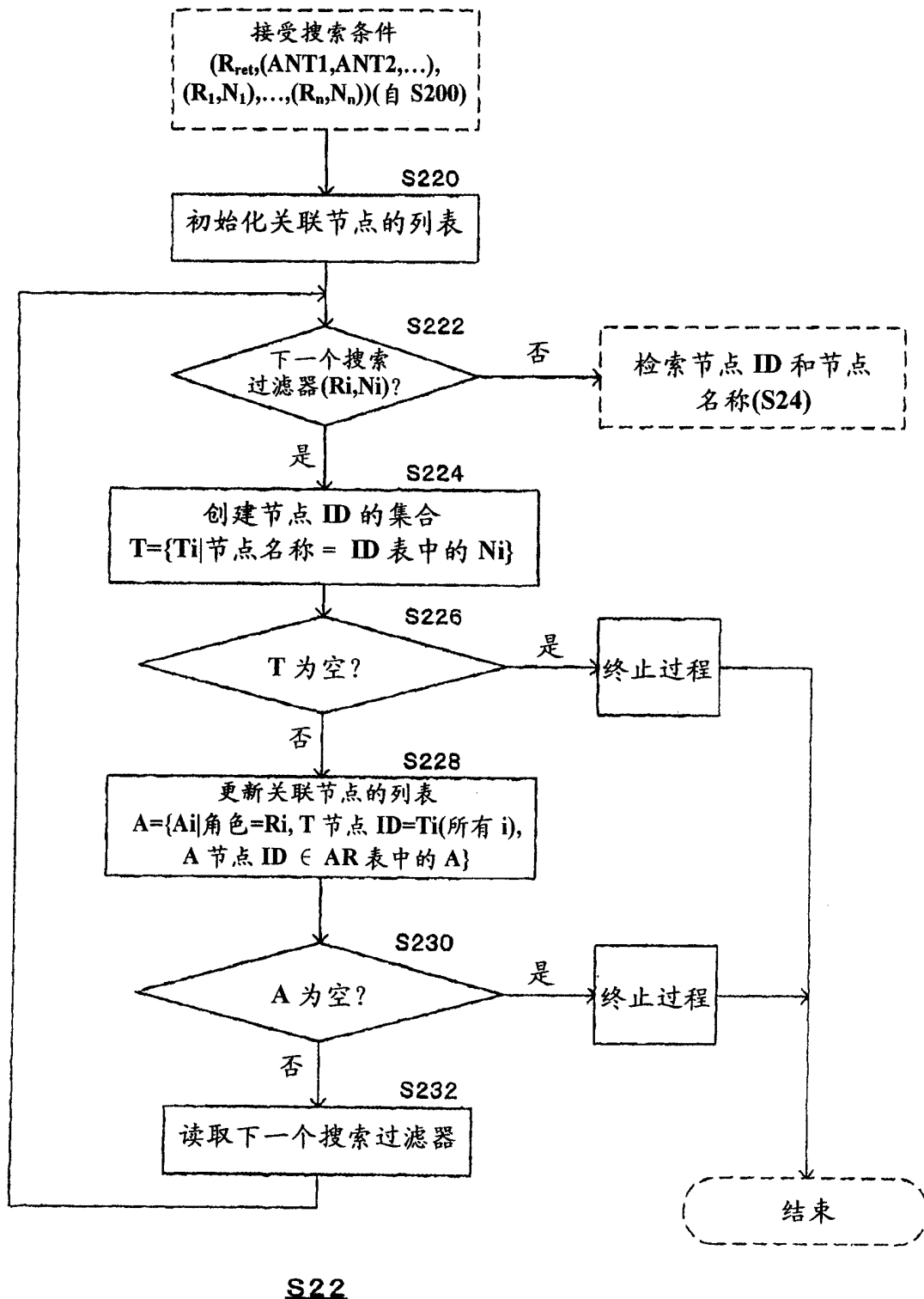


图 17

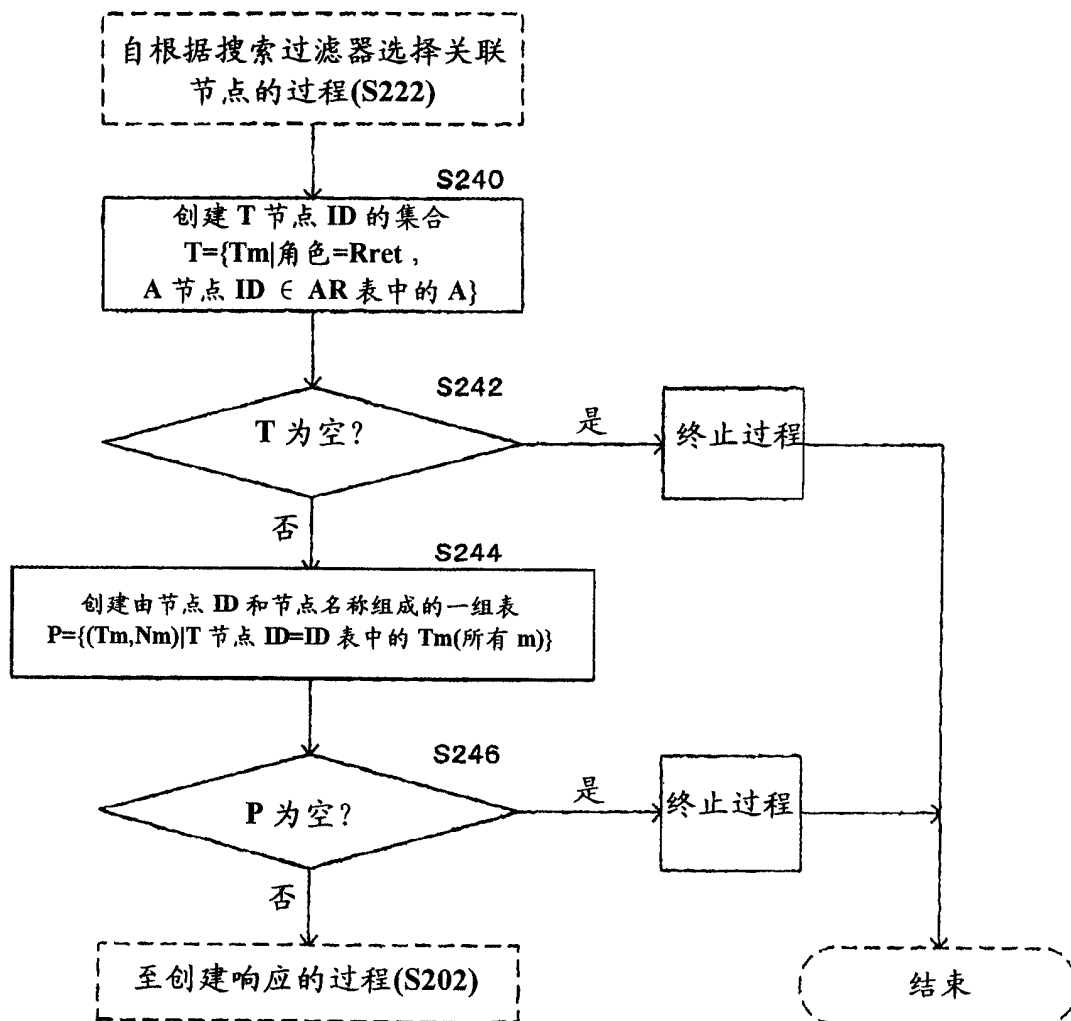
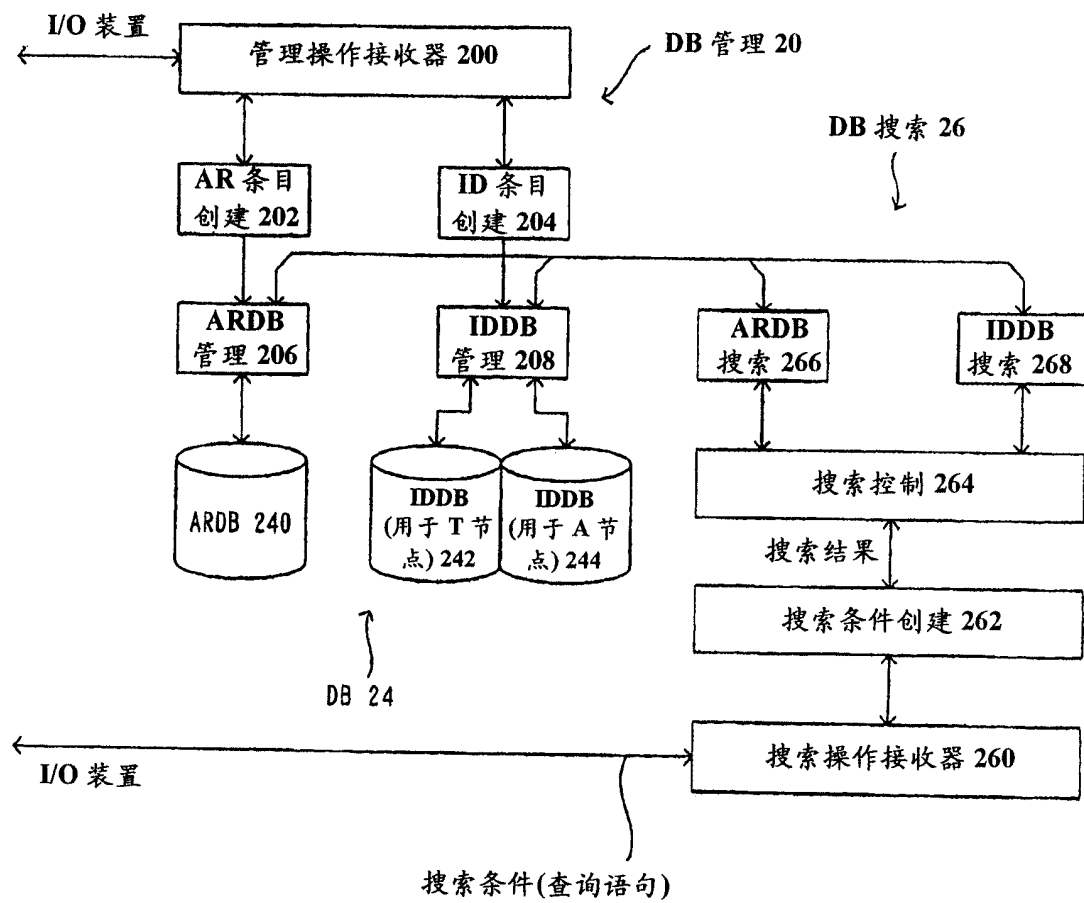
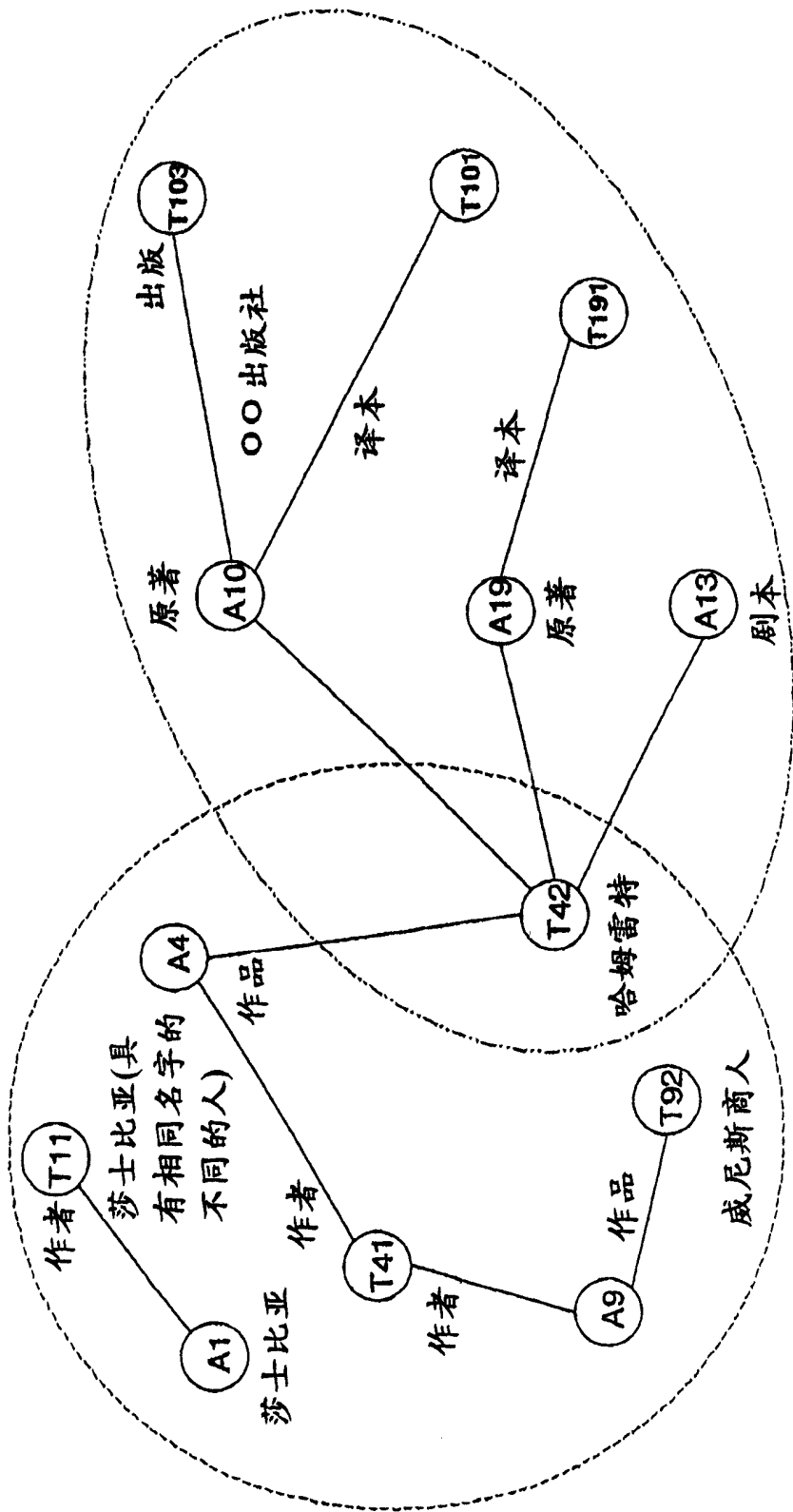
S24

图 18



DB 程序 2

图 19



第一步骤
第二步骤

查询语句的实例“查找出版剧作家莎士比亚作为作者写作的作品之一 - 戏剧哈姆雷特的原著的译本的出版商的名称”

图 20

AR 表

A 节点	T 节点	关联角色
A1	T11	作者
A4	T41	作者
A4	T42	作品
A9	T41	作者
A9	T92	作品
A10	T42	原著
A10	T101	译本
A10	T103	出版
A13	T42	剧本
A19	T42	原著
A19	T191	译本

图 21

ID 表(用于 T 节点)

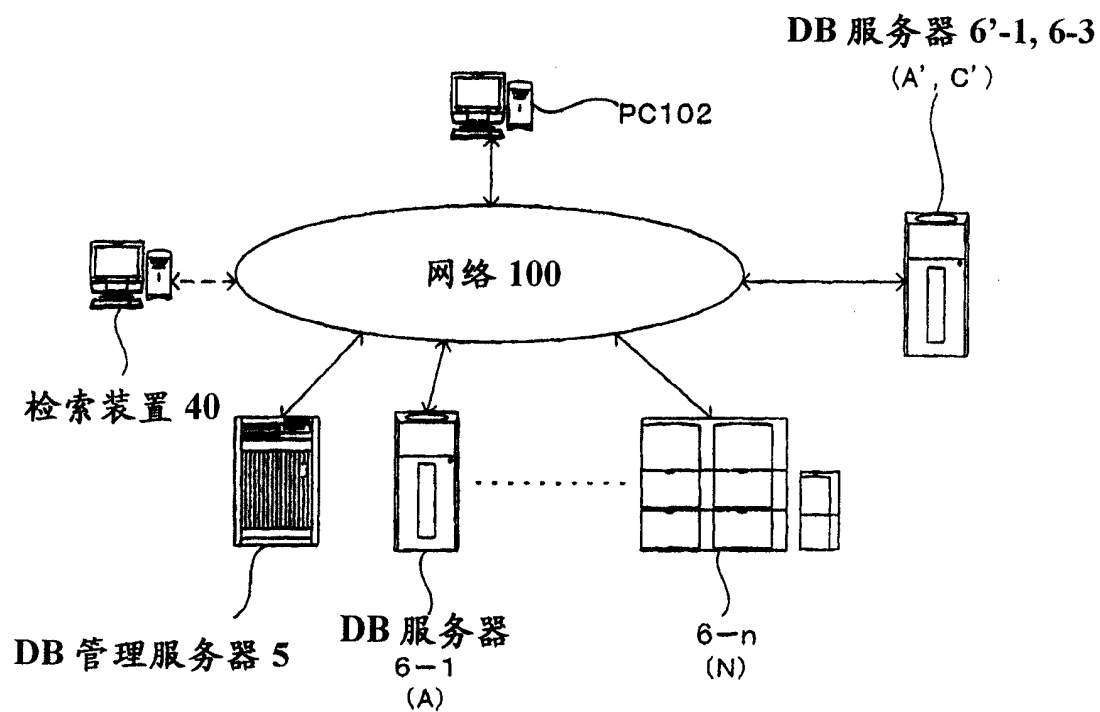
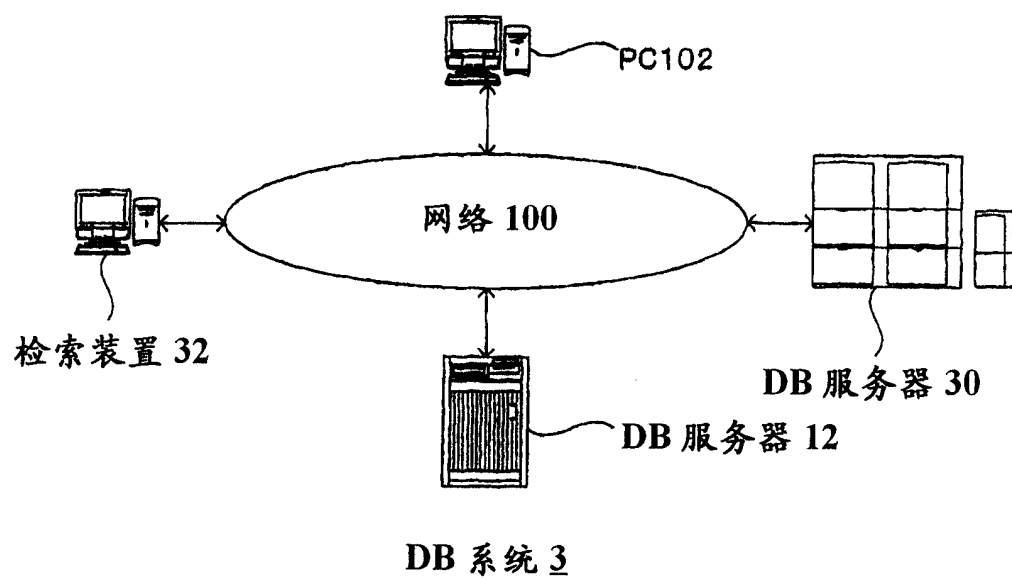
节点 ID	节点类型	节点名称
T11	剧作者	莎士比亚
T31	美国人	莎士比亚
T41	剧作者	莎士比亚
T42	戏剧	哈姆雷特
T51	店主	莎士比亚
T81	地点	莎士比亚
T92	戏剧	威尼斯商人
:	:	:
T101	戏剧	哈姆雷特的日译本
T103	出版商	OO 出版社
T191	戏剧	哈姆雷特的中译本

图 22

ID 表(用于 A 节点)

节点 ID	节点类型	节点名称
A1	关于出生的信息	(空)
A4	关于写作的信息	(空)
A9	关于写作的信息	(空)
A10	关于译本的信息	(空)
A13	关于演出的信息	(空)
A19	关于译本的信息	(空)
:	:	:

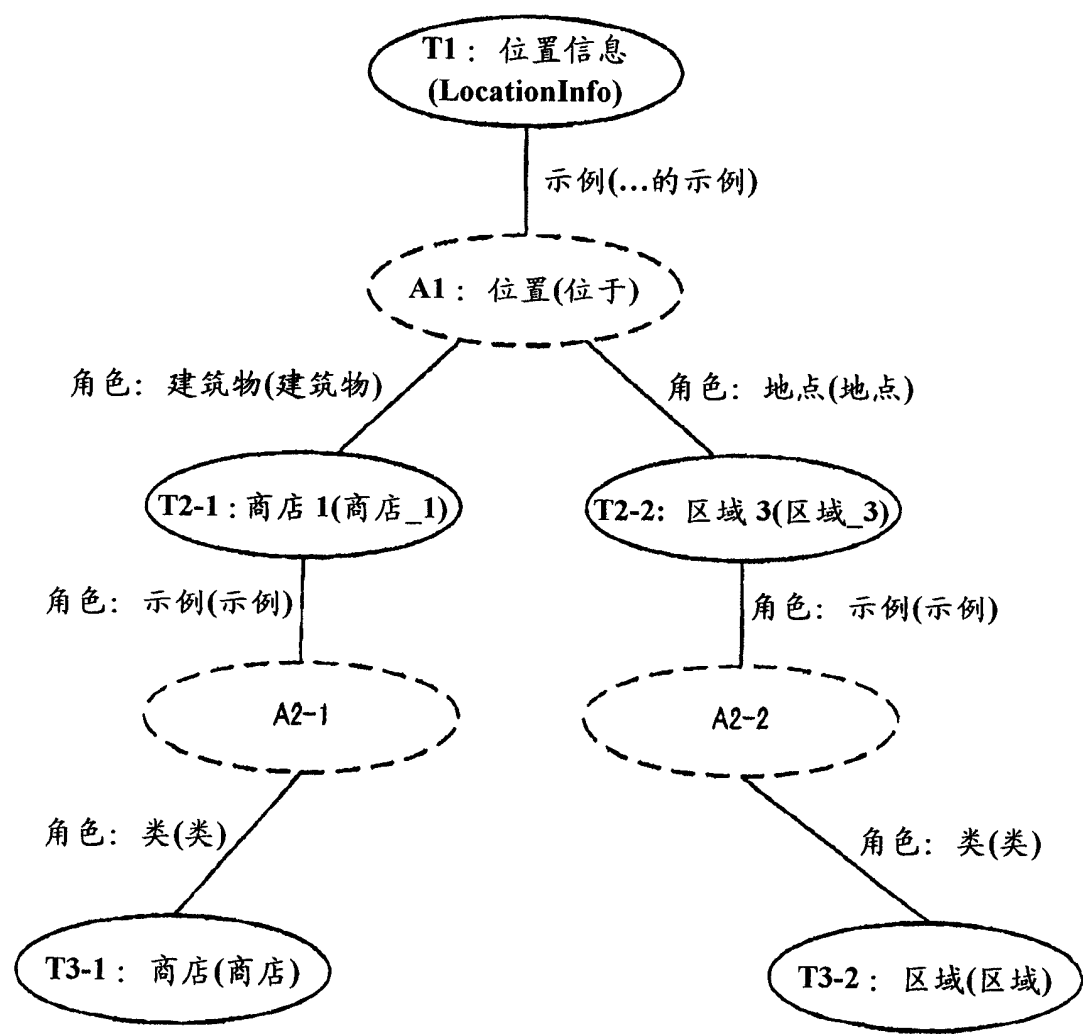
图 23



DB 系统 4

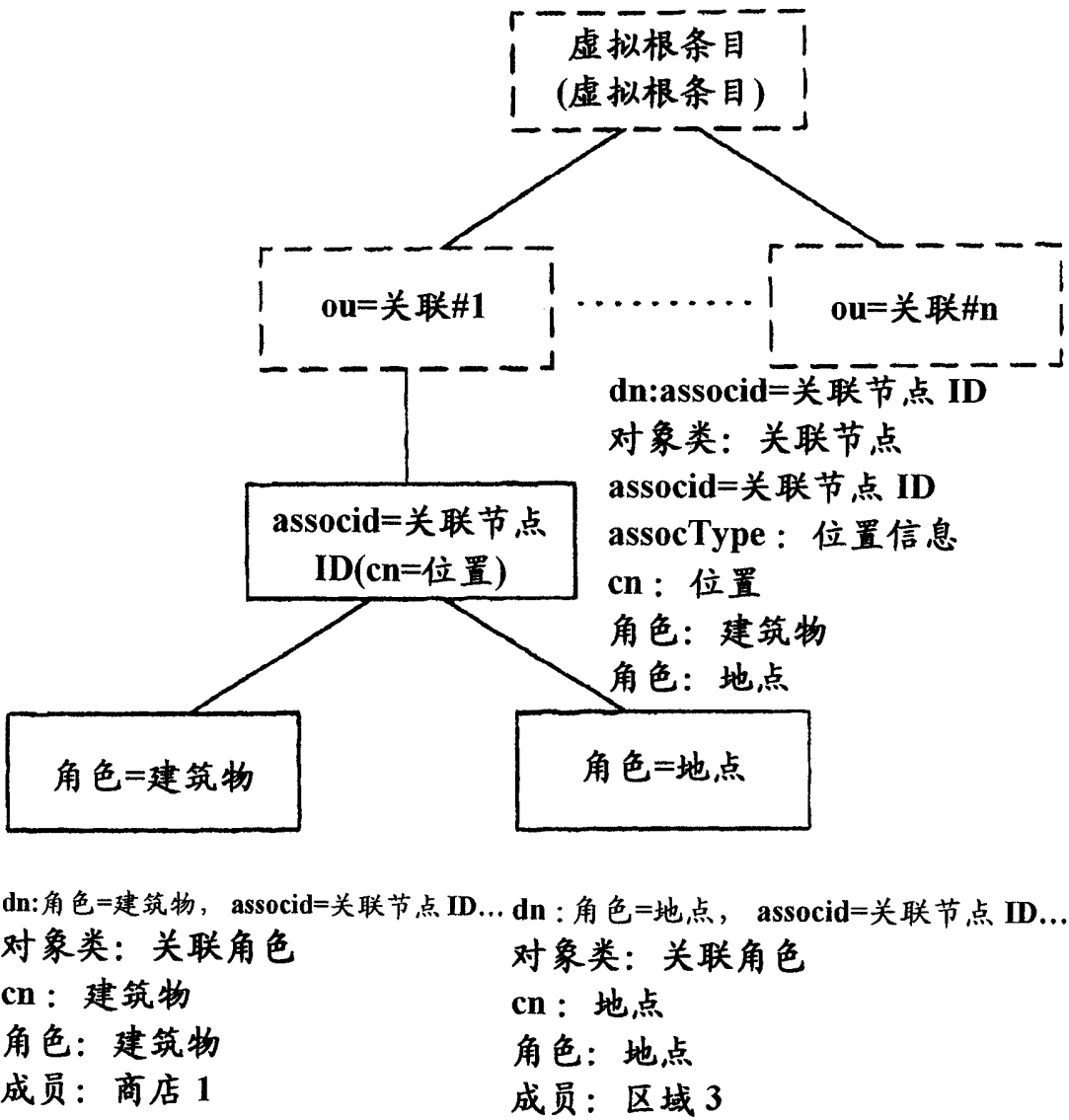
图 24

图 25



相关数据的图形表示

图 26



相关数据的 DIT 表示

图 27

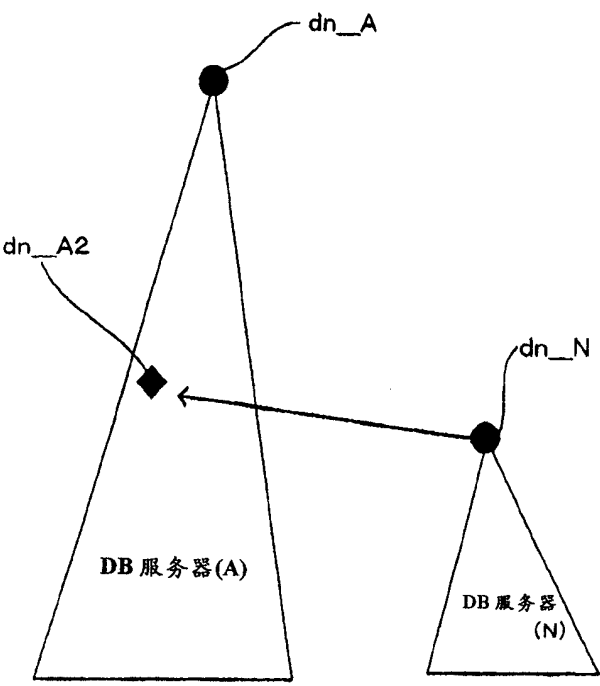
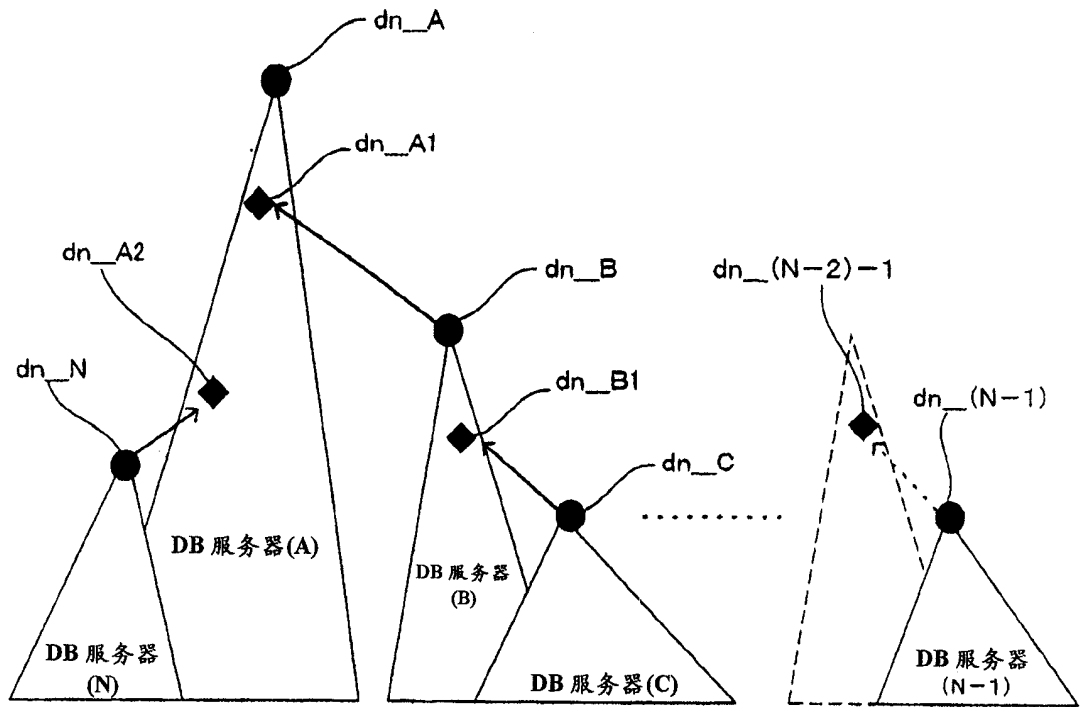


图 28

图 29

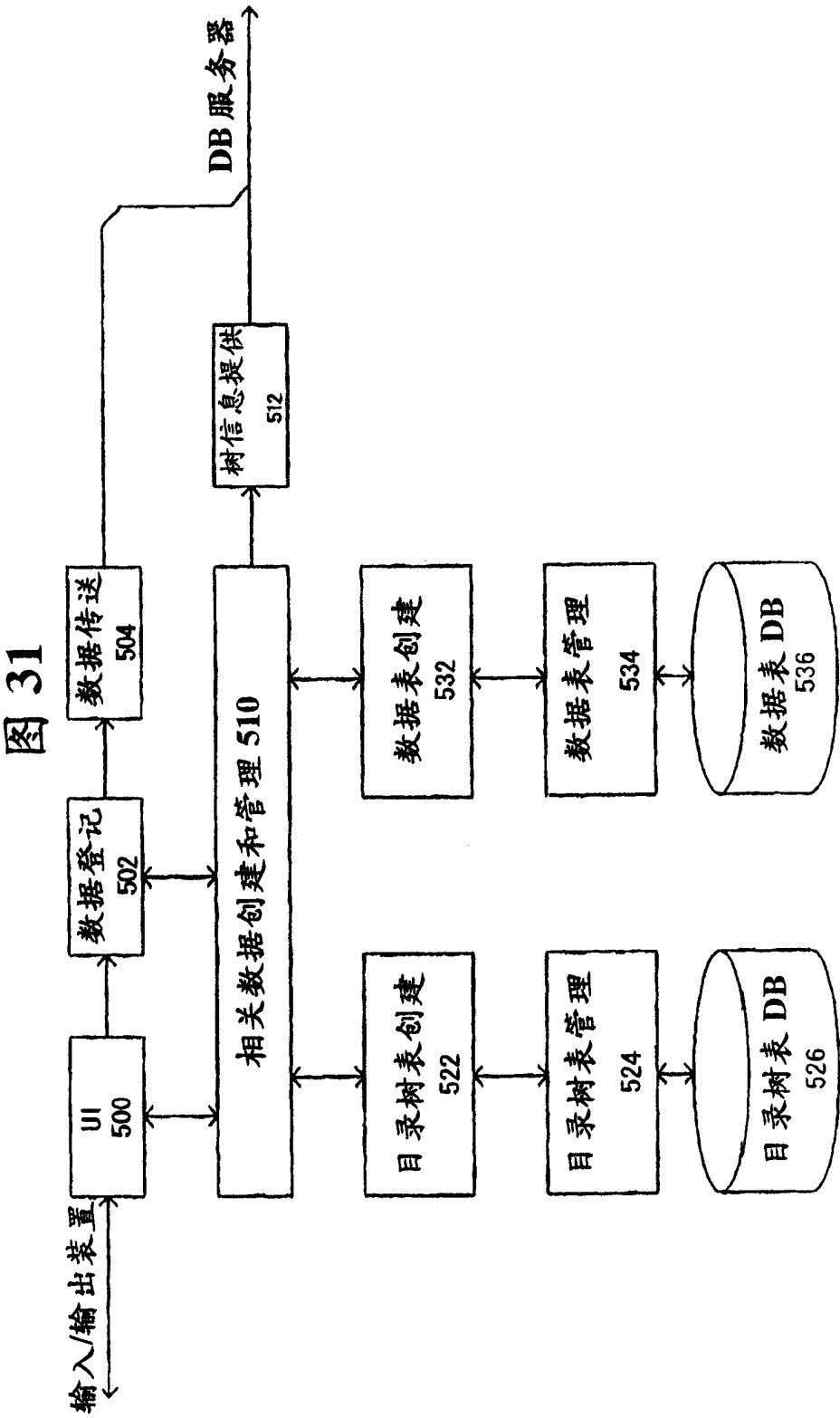
DB 服务器名称	顶部条目	引用条目	复制服务器
A	dn_A	N/A	A'
B	dn_B	dn_A-1	N/A
C	dn_C	dn_B-1	C'
D	dn_D	dn_A-2	N/A
N	dn_N	dn_(N-1)-1	N/A

目录树表

子树	分类类型	分类值	顶部条目
a	类别(类别)	附属公司信息(附属)	dn_A
b	位置信息 (LocationInfo)	位置(位于)	dn_B
n-1	建筑物 (建筑物)	商店 1 (商店_1)	dn_(N-1)
n			

数据表

图 30



DB 管理程序 50

图 32

莎士比亚的作品

数据输入关于 ~

关联名称:

关于哈姆雷特

关联类型:

关于写作的信息▼

其它:

名称:

莎士比亚

角色:

作者▼

其它:

添加

清除

退出

图 33

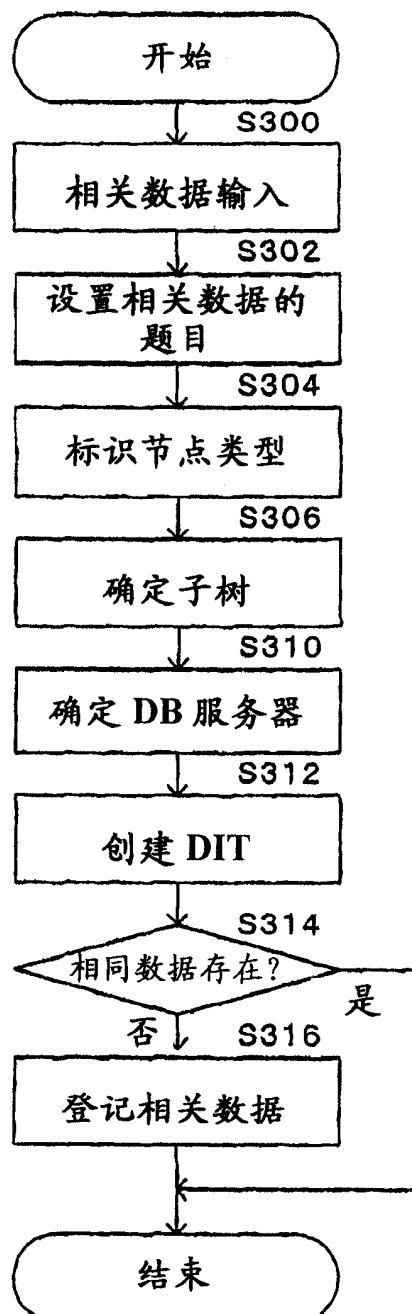
S30

图 34

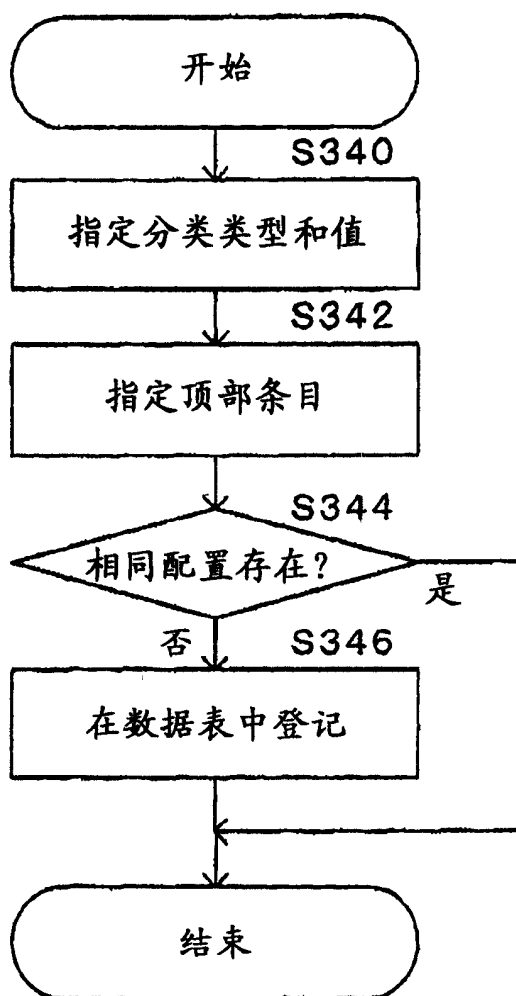
**S34**

图 35

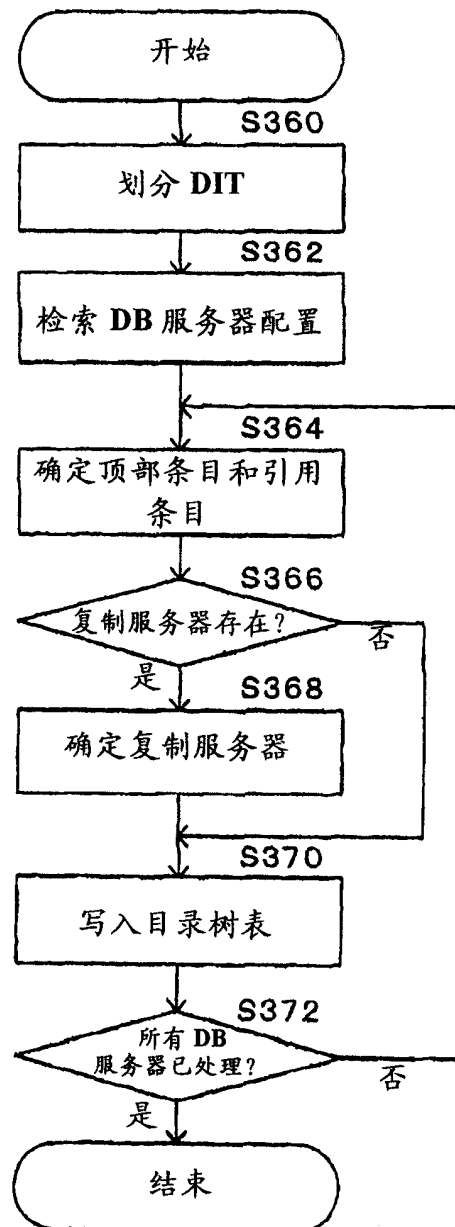
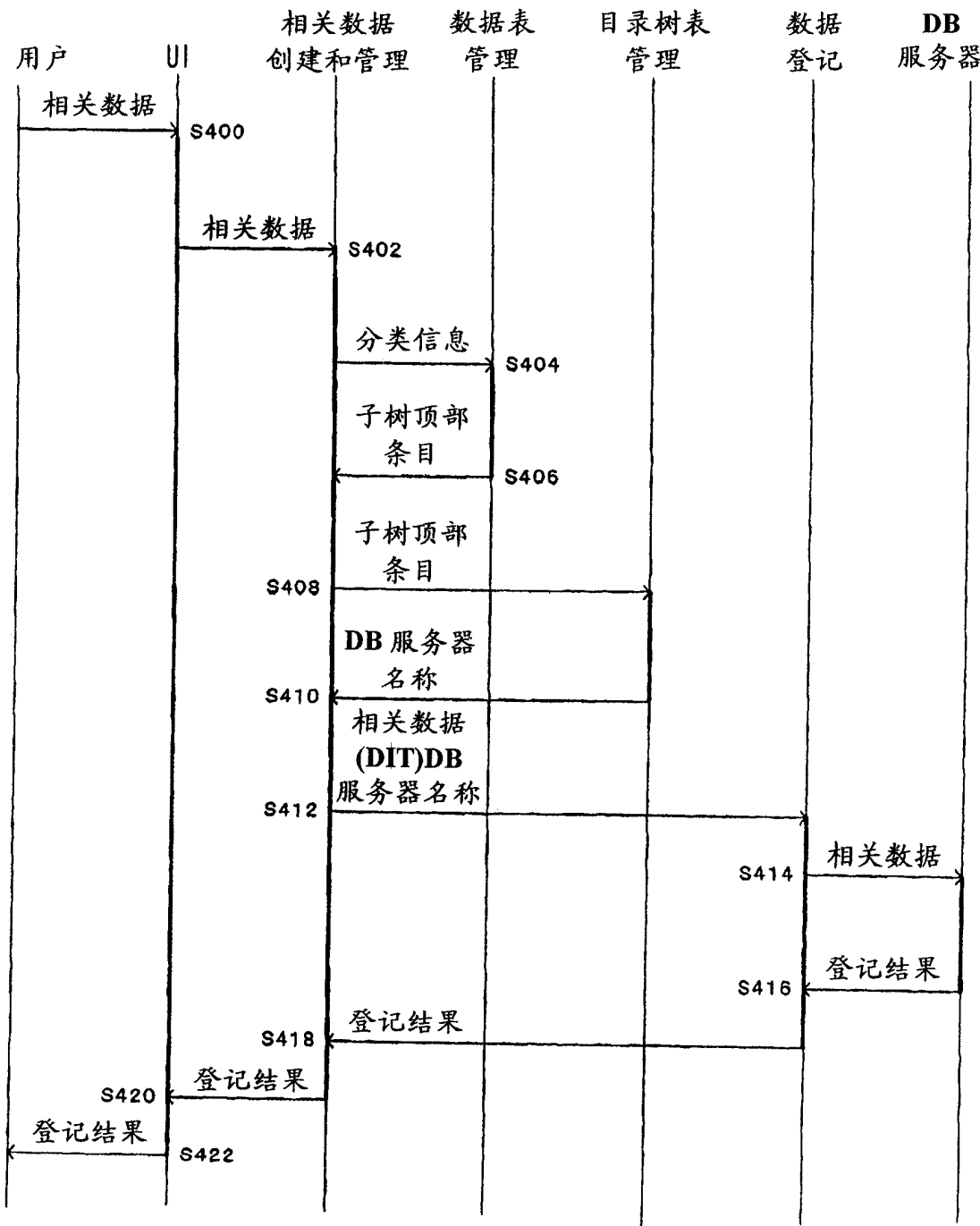
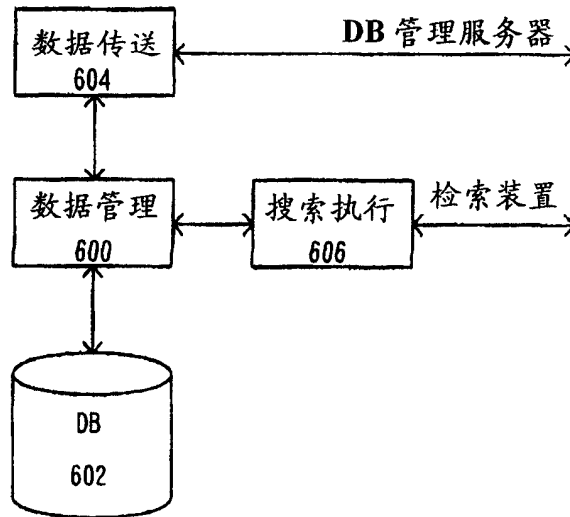
**S36**

图 36

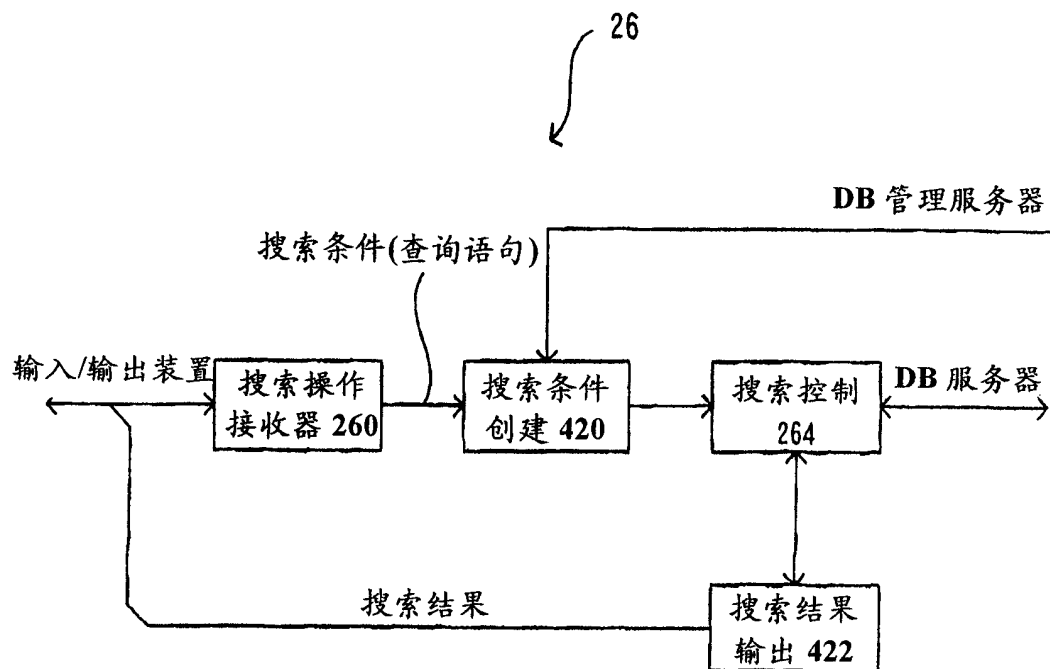


S40

图 37



DB 程序 60



搜索程序 42

图 38

图 39

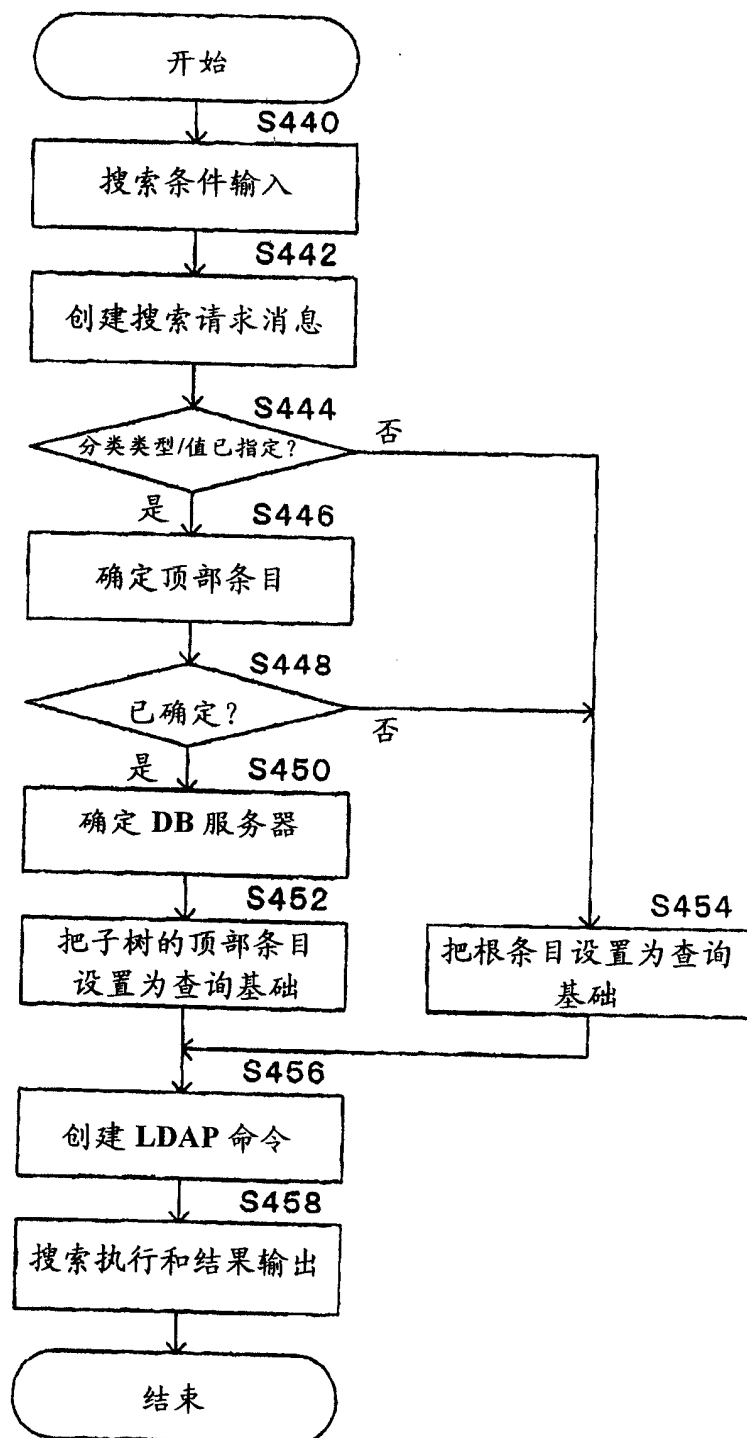
**S44**

图 40

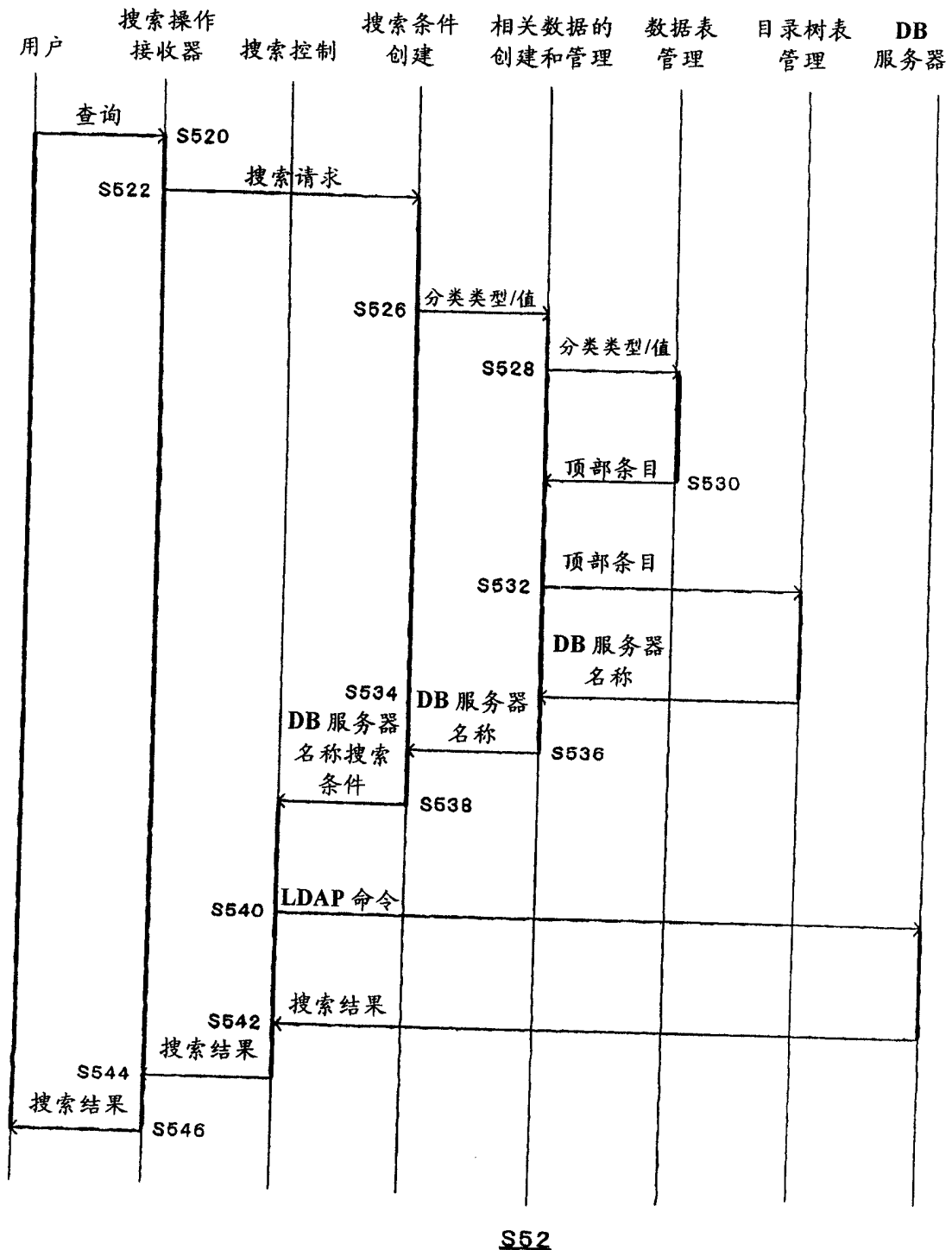


图 41

平面目录结构

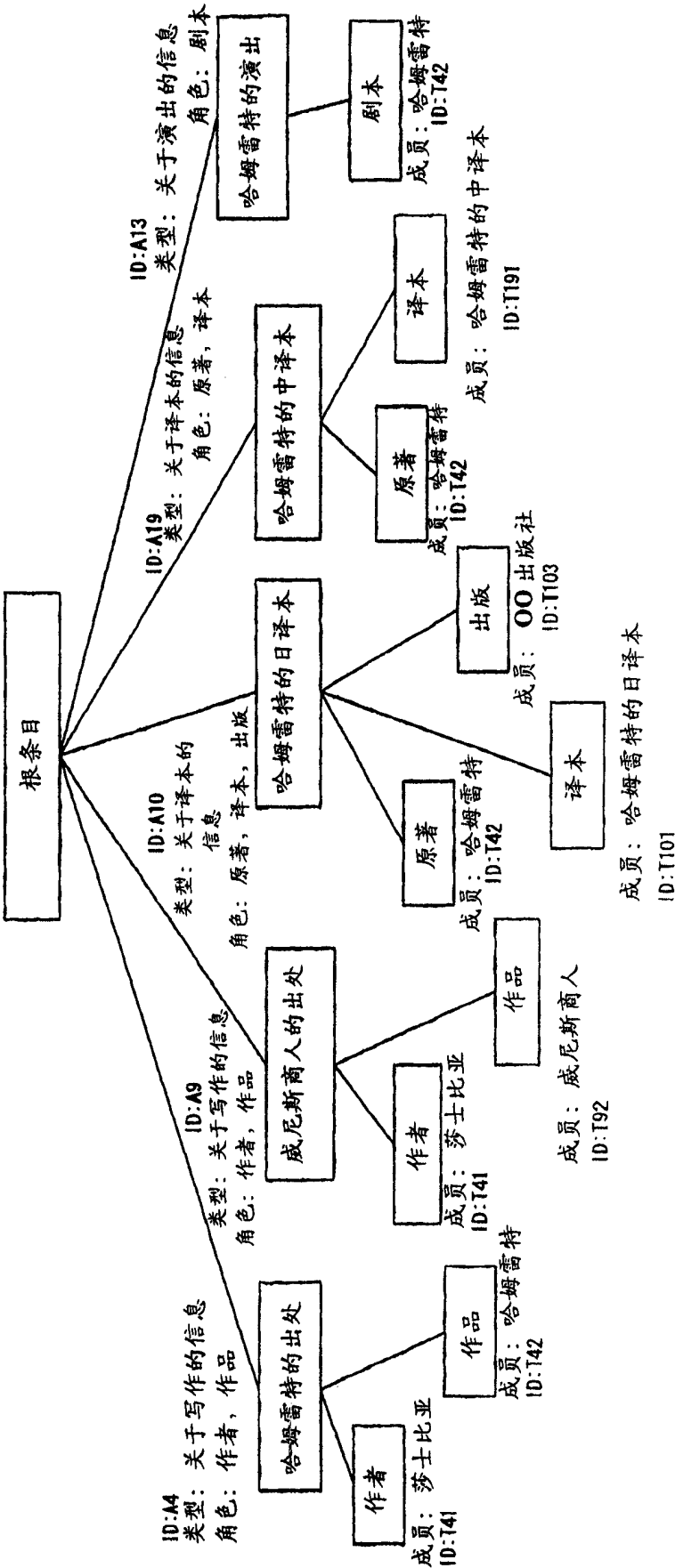


图 42

基于题目的分层结构

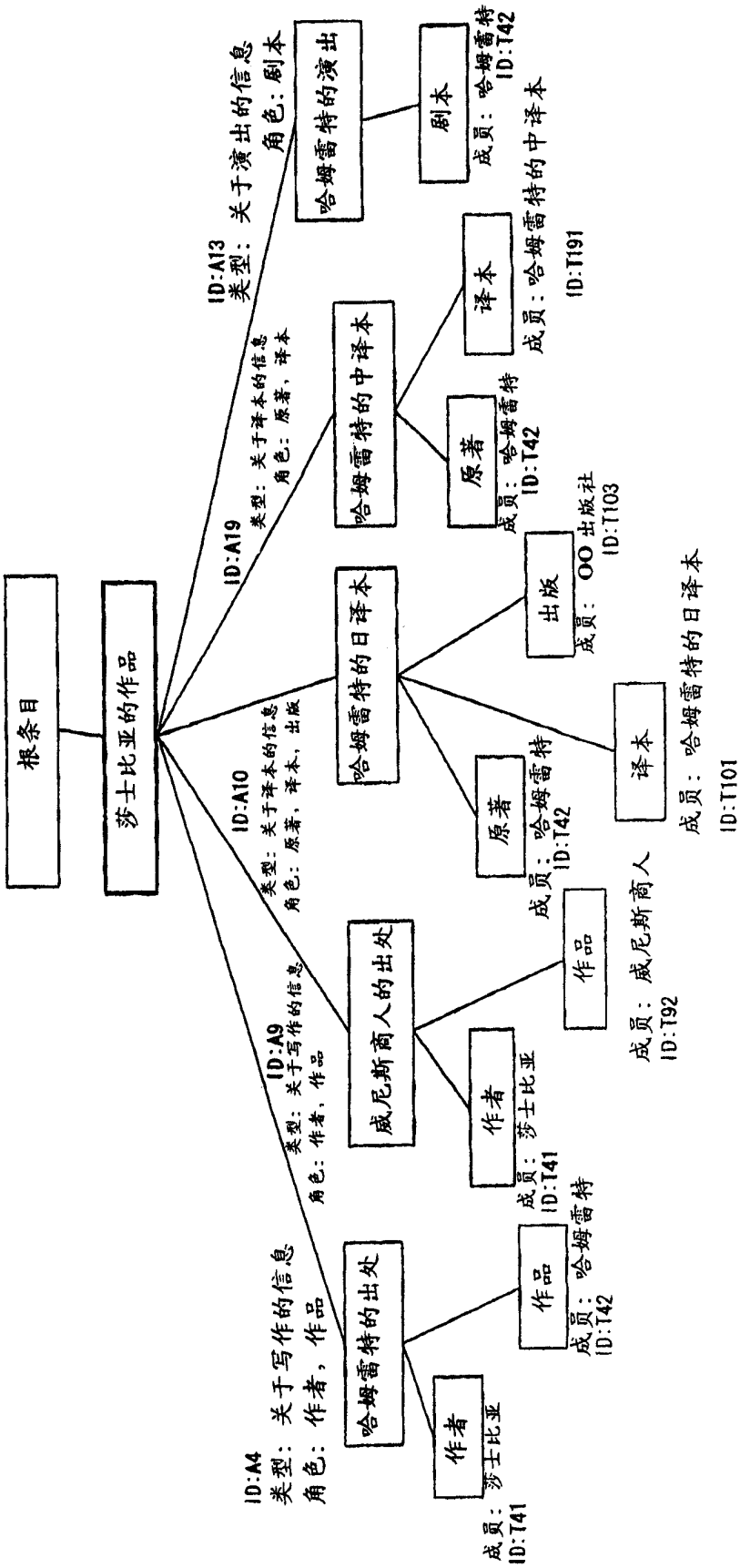


图 43
基于关联类型的分层结构

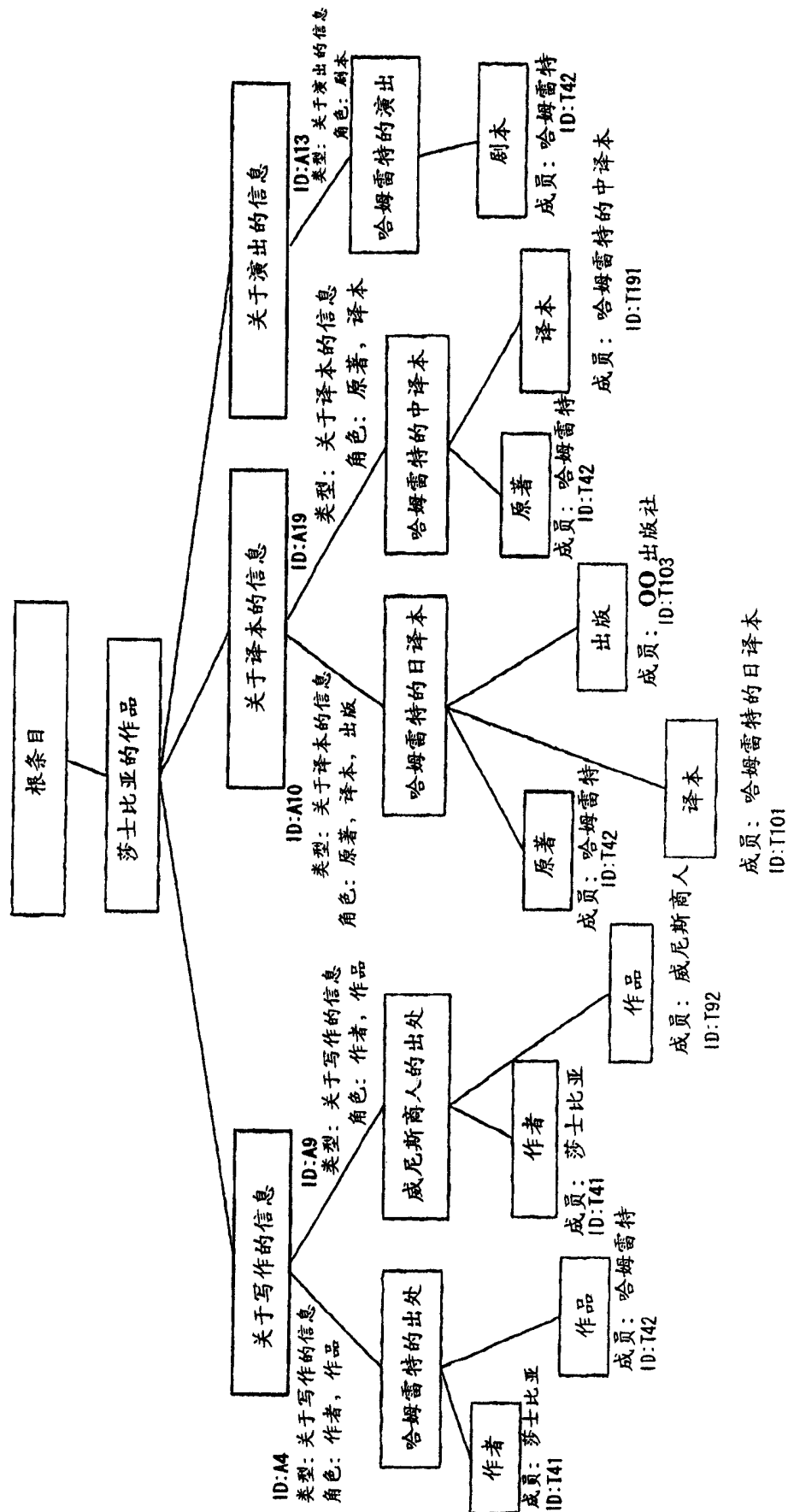


图 44

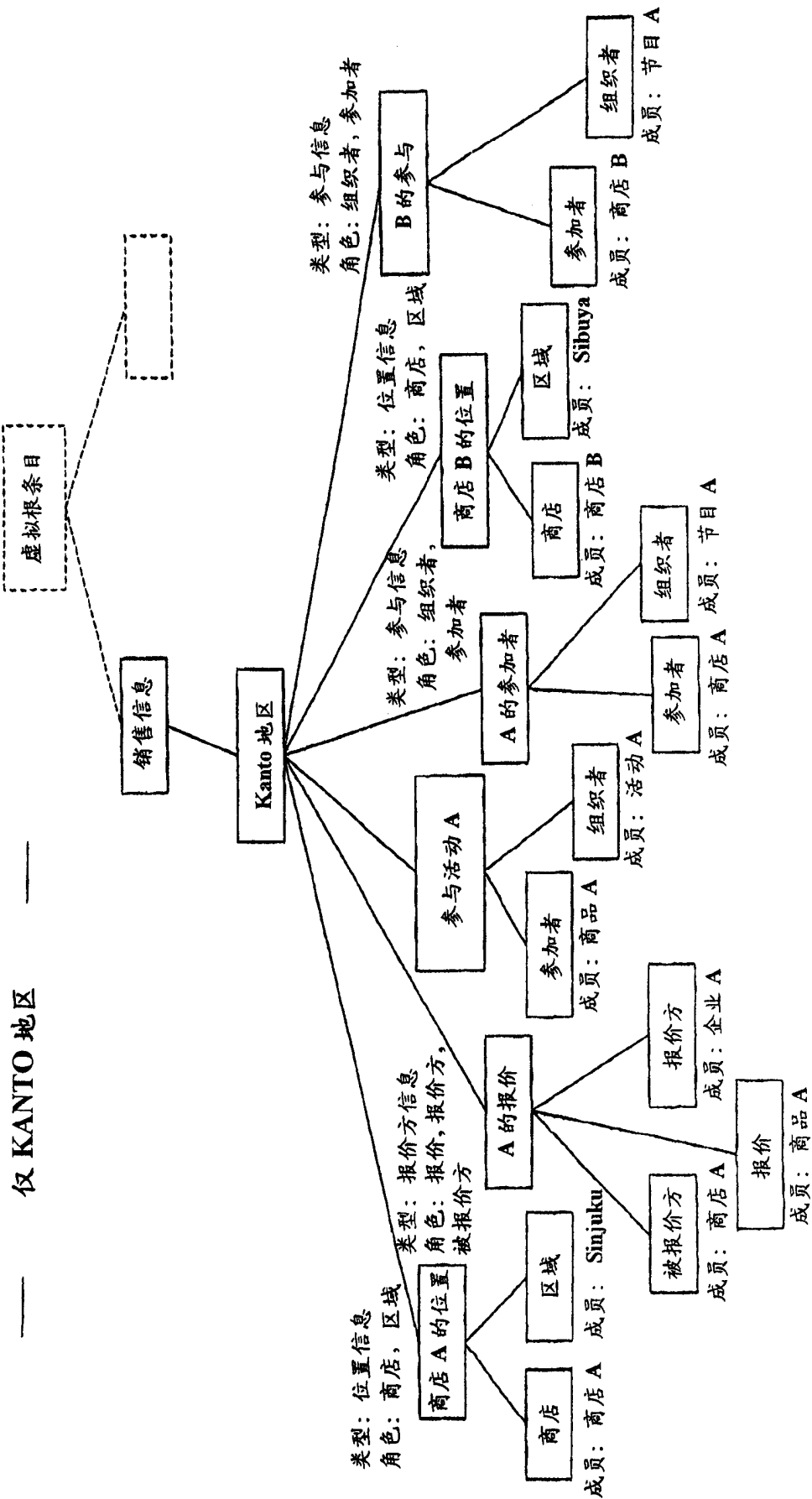
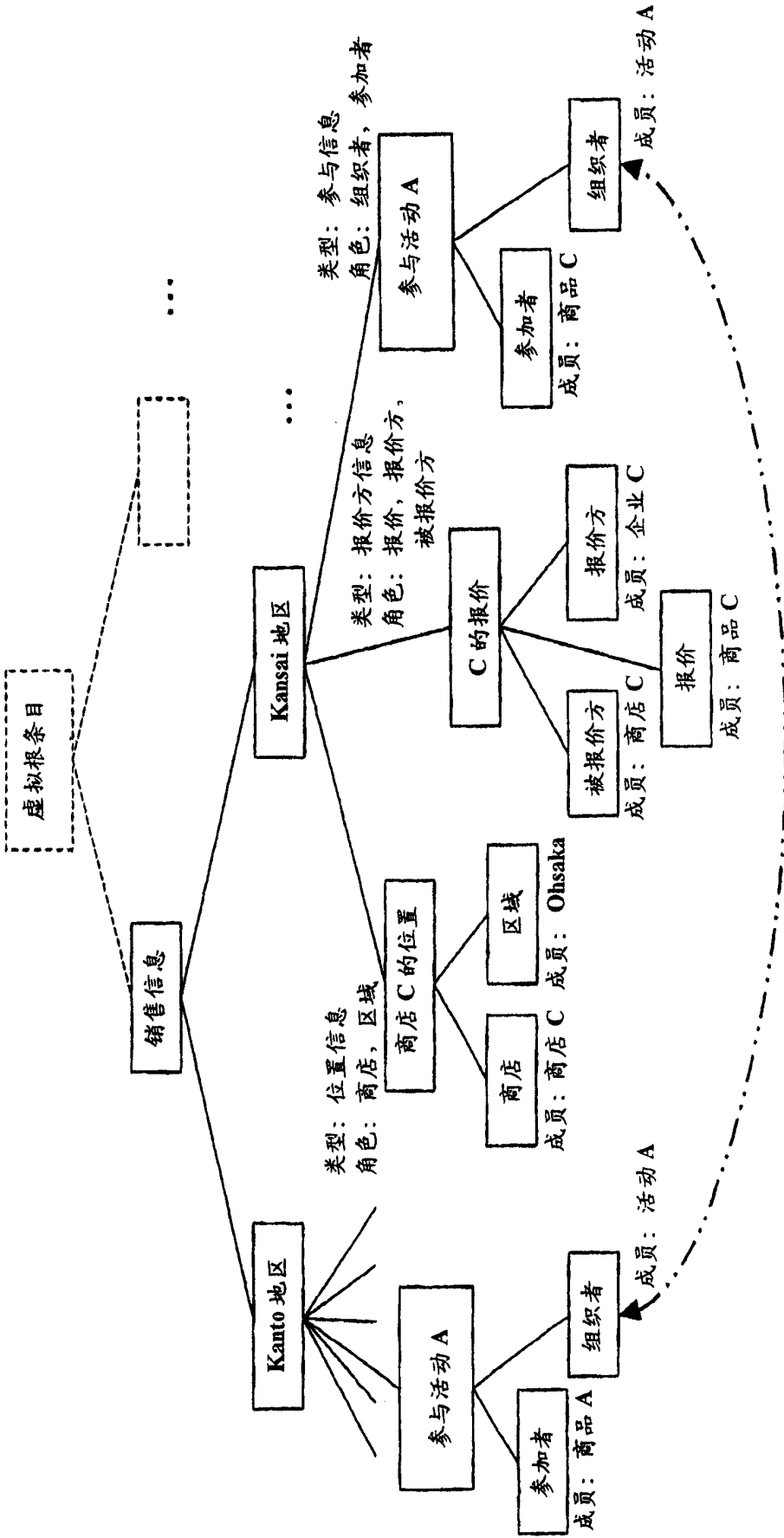


图 45



跨地区组织的销售活动

图 46

服务器名称	顶部条目	引用条目	复制服务器
服务器 A	dn_A	N/A	服务器 A'
服务器 B	dn_B	dn_A ₁	N/A
服务器 C	dn_C	dn_B ₁	服务器 C'
:	:	:	:

图 47

子树	分类类型	分类值	顶部条目
a	类别	附属公司信息	dn_A
b	地区	Kanto 地区	dn_B
:	:	:	:

图 48

Kanto 地区

搜索~中的销售信息

参与信息

▼

角色

组织者

▼

搜索成员

搜索过滤器:

名称

商店 B

角色

成员

▼

设置(添加)

清除

退出

图 49

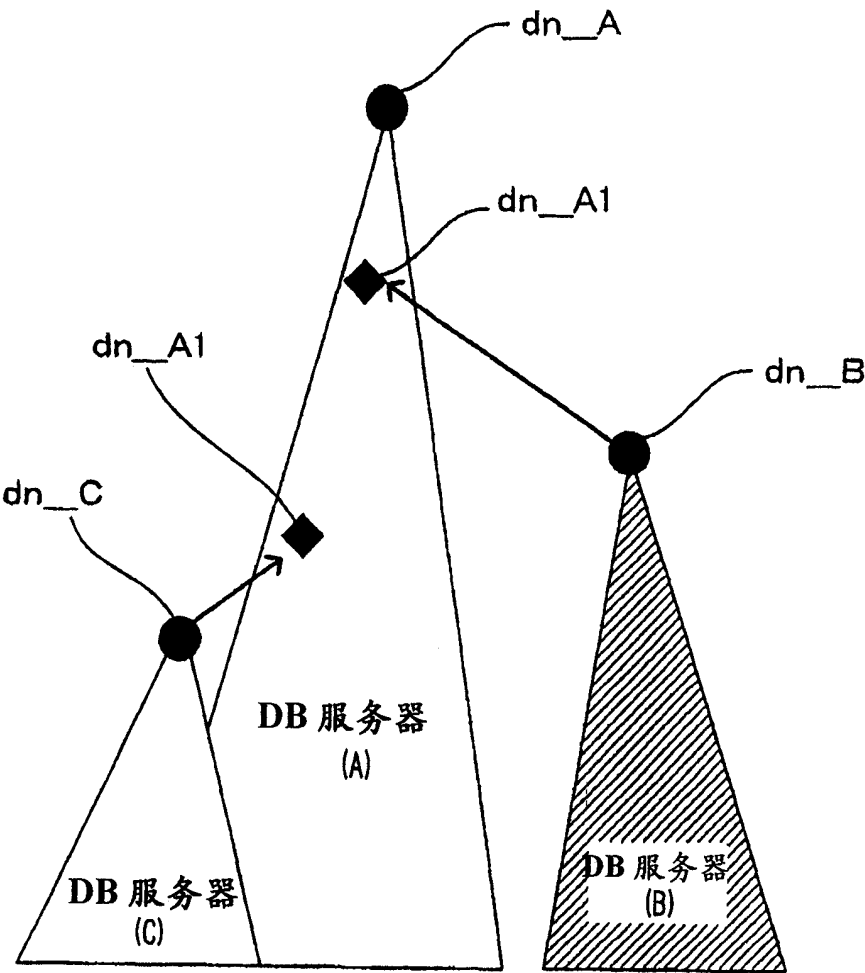


图 50 <searchRequest>
 <condition region=“ Kanto 地区” category= “销售信息” />
 <query result=“res1”>
 <associationType> 参与信息 </associationType>
 <targetRole> 组织者 </target Role>
 <memberSpec>
 <memberID> 商店 B </memberID>
 <role> 参加者 </role>
 </memberSpec>
 </query>
 <query result=“res2”>
 <associationType> 参与信息 </associationType>
 <targetRole> 参加者 </targetRole>
 <memberSpec>
 <memberID>\$res1;</memberID>
 <role> 组织者 </role>
 </memberSpec>
 </query>
 </searchRequest>

图 51

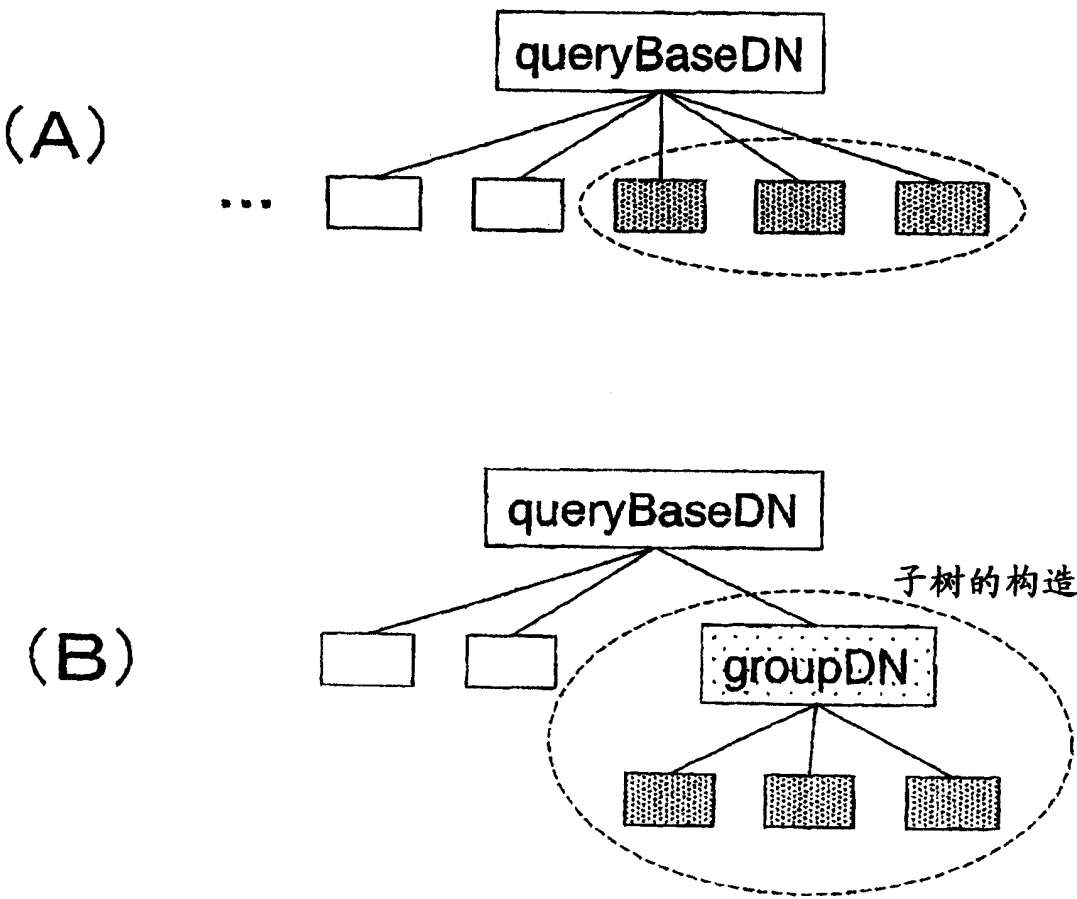


图 52

搜索条件管理表

查询基础	过滤器	新的查询基础
queryBaseDN	queryFilter	groupDN
▪ ▪ ▪	▪ ▪ ▪	▪ ▪ ▪

图 53

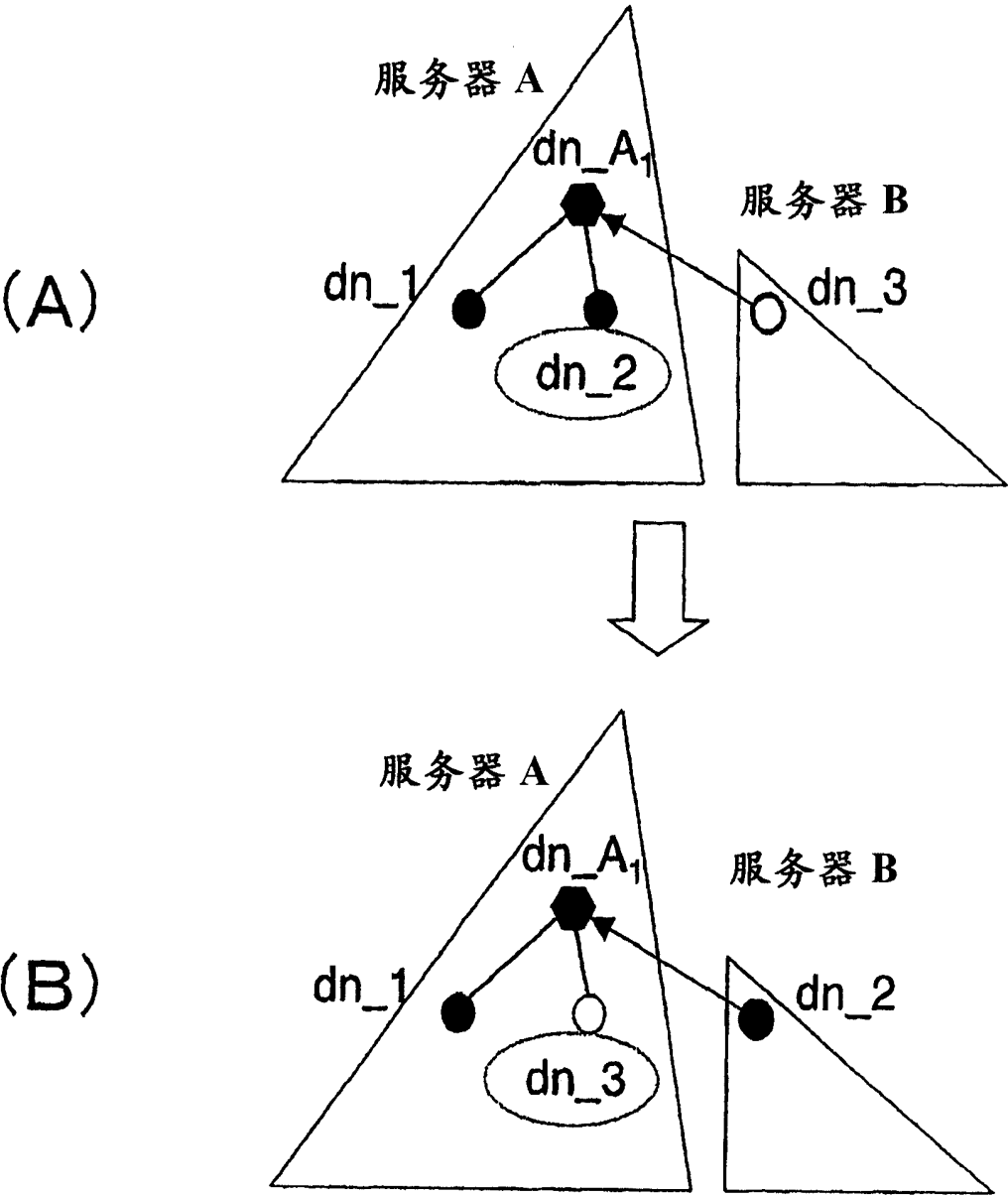


图 54

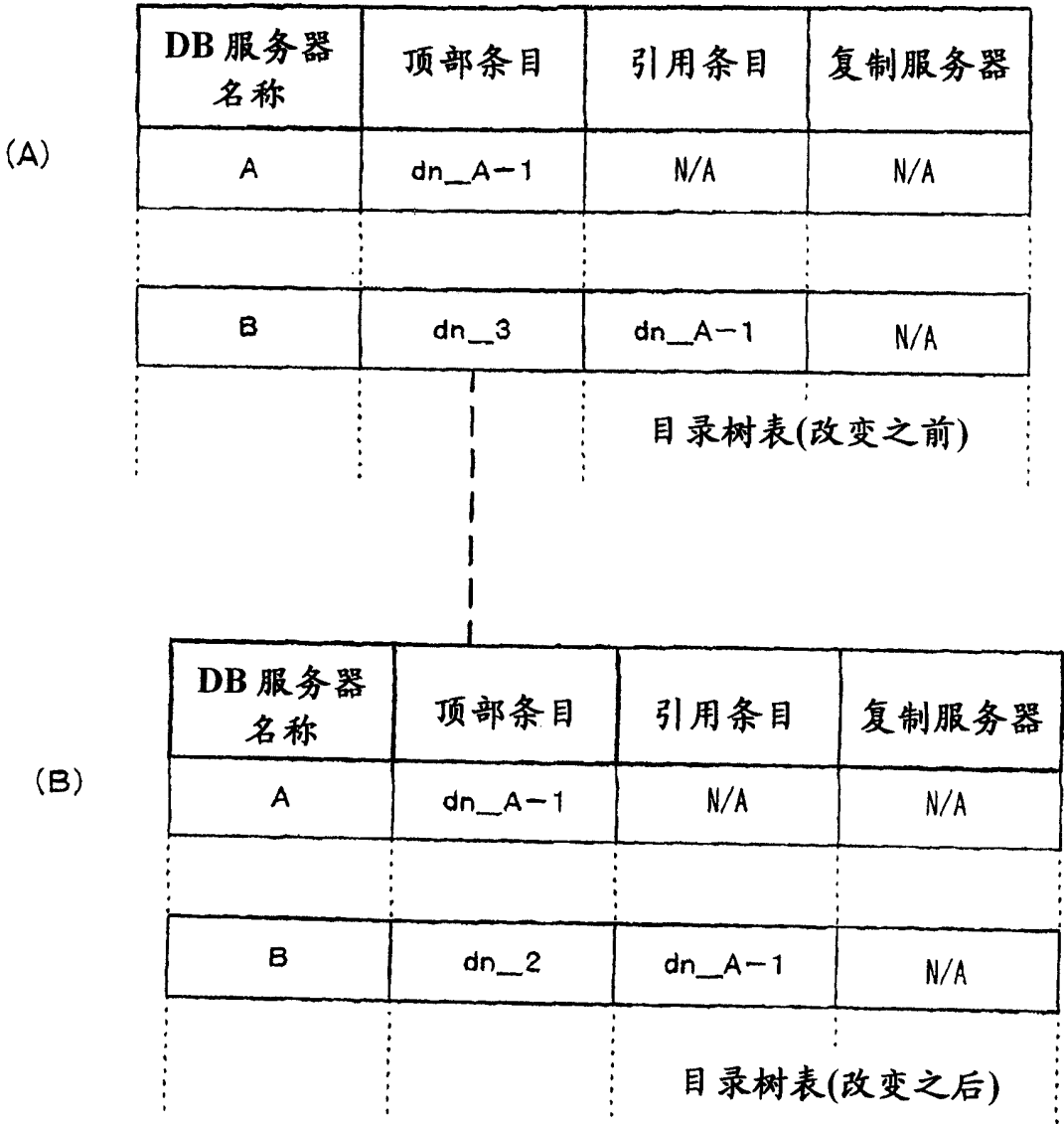
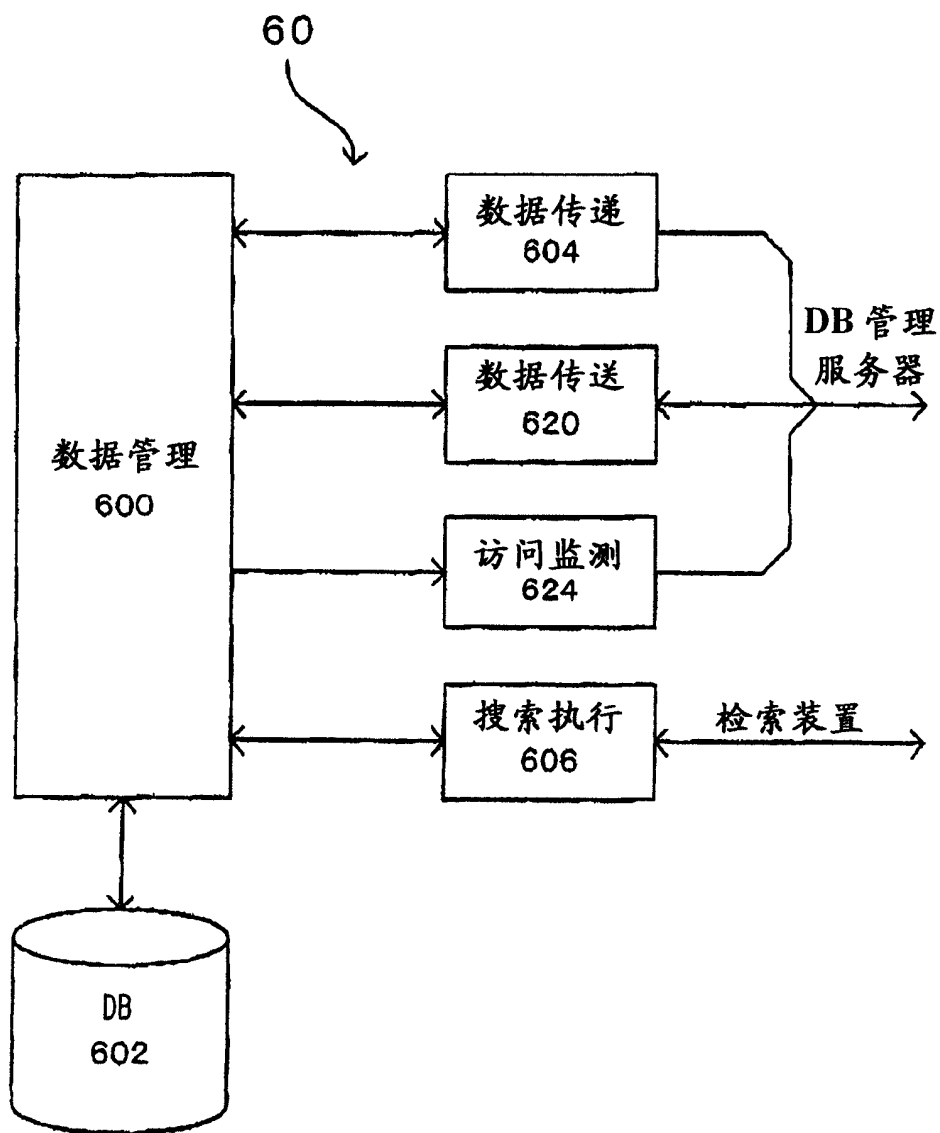
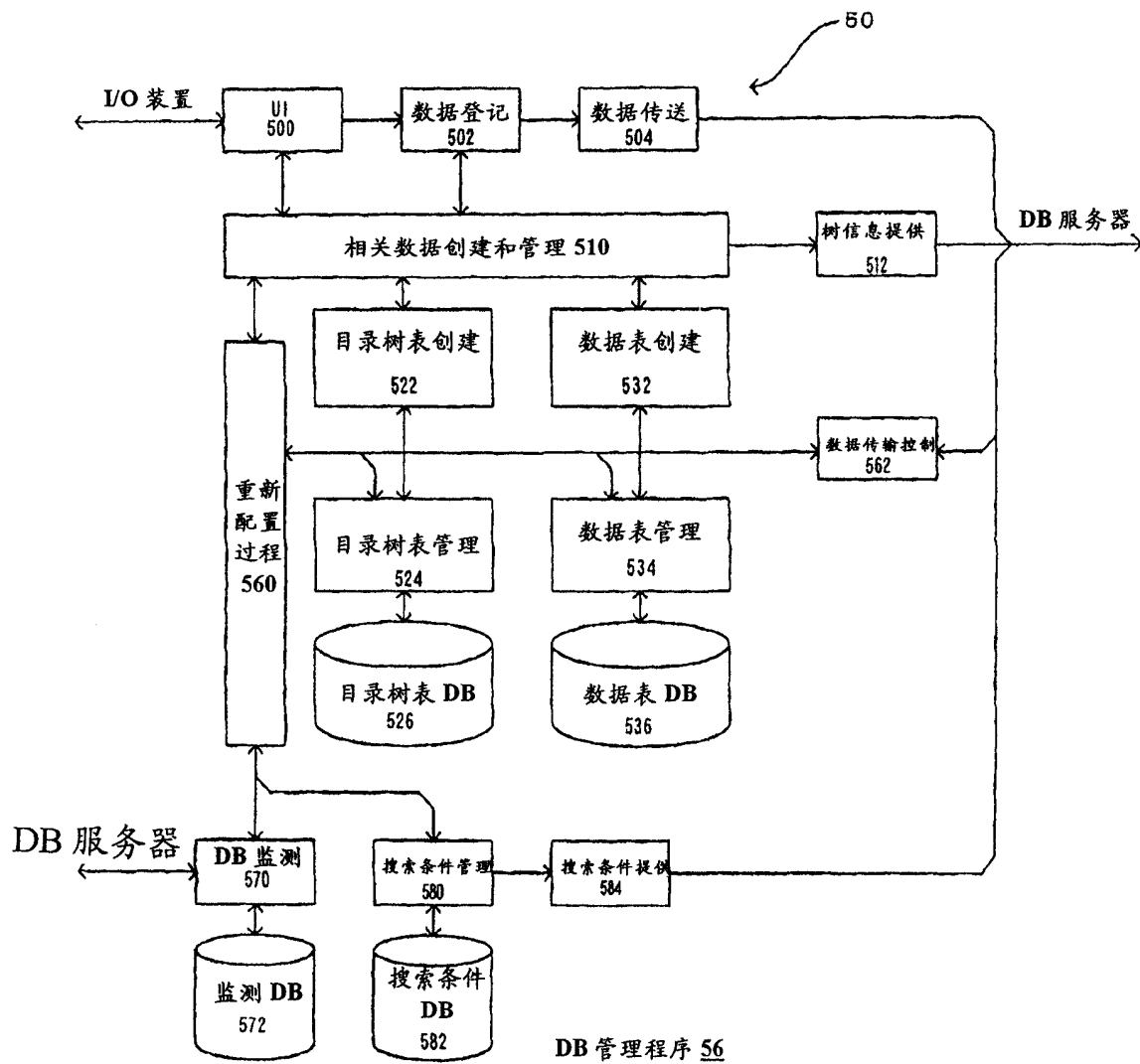


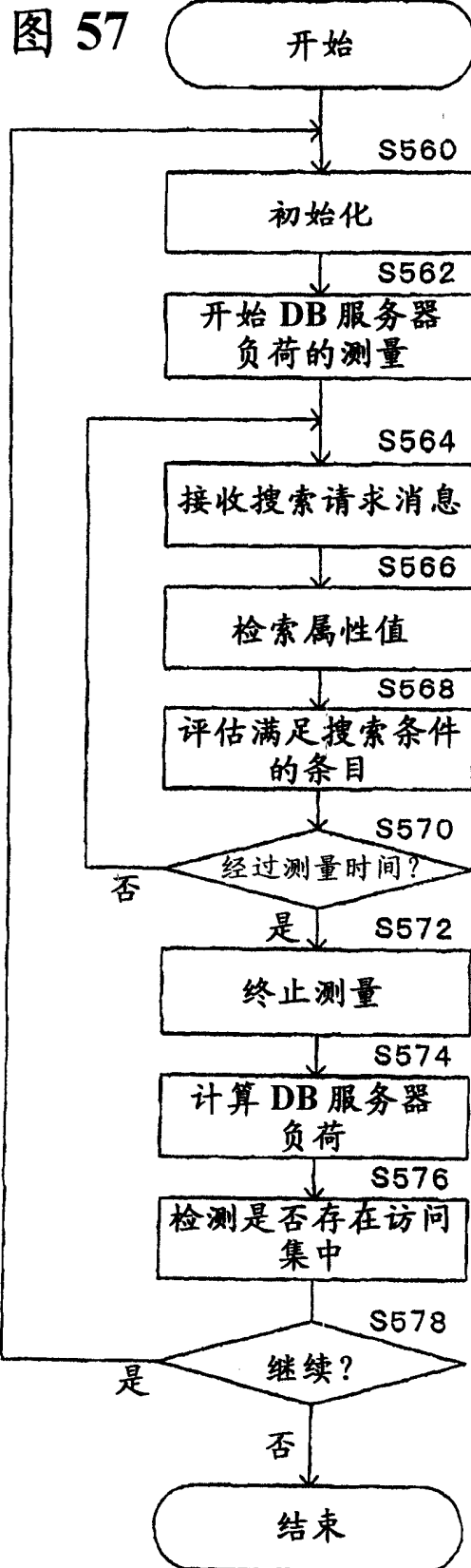
图 55



DB 程序 62

图 56





S56

图 58

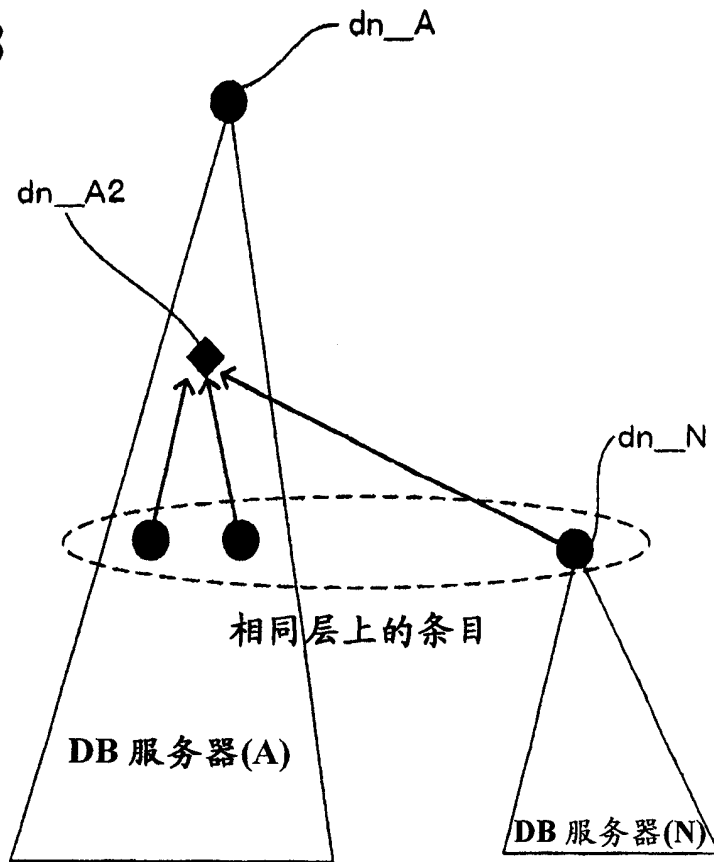


图 59

