



(12) 发明专利

(10) 授权公告号 CN 101371220 B

(45) 授权公告日 2010. 11. 10

(21) 申请号 200780002228. 2

代理人 余刚 尚志峰

(22) 申请日 2007. 01. 10

(51) Int. Cl.

(30) 优先权数据

G06F 7/00 (2006. 01)

60/758, 249 2006. 01. 10 US

11/651, 739 2007. 01. 09 US

(85) PCT申请进入国家阶段日

2008. 07. 10

(86) PCT申请的申请数据

PCT/US2007/000664 2007. 01. 10

(87) PCT申请的公布数据

W02007/082005 EN 2007. 07. 19

(73) 专利权人 UNZ. ORG 有限责任公司

地址 美国加利福尼亚州

(72) 发明人 罗恩·K·温兹

(74) 专利代理机构 北京康信知识产权代理有限公司 11240

(56) 对比文件

CN 1647076 A, 2005. 07. 27, 全文.

US 6199098 B1, 2001. 03. 06, 全文.

US 2001/0051949 A1, 2001. 12. 13, 全文.

US 2005/0114763 A1, 2005. 05. 26, 全文.

审查员 陈安安

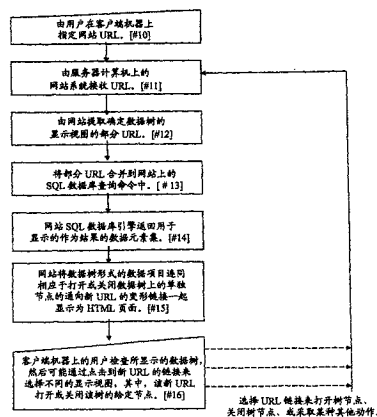
权利要求书 3 页 说明书 6 页 附图 6 页

(54) 发明名称

用于传送和显示分层数据的设备以及机器实现方法

(57) 摘要

一种用于传送和显示分层数据的设备包括：数据储存器，连接至一个或多个处理器，以及包含在数据库结构中并被组织为分层数据树的多个数据元素，以及执行接收来自客户端计算机的网络位置标识符的指令，其中，所述网络位置标识符至少包括指定数据树的第一视图的部分；利用网络位置标识符的该部分来创建关系数据库查询并将该查询提交给关系数据库；接收来自关系数据库的与第一视图相关联的数据元素集；生成包括该数据元素集的第一电子文档，包括对应于对数据树中的节点执行动作而生成通向其他网络位置标识符的变形链接；以及将第一电子文档提供给客户端计算机。



1. 一种用于传送和显示分层数据的设备,包括:

一个或多个处理器;

数据储存器,连接至一个或多个所述处理器,在所述数据储存器上记录有多个数据元素,所述多个数据元素包含在关系数据库结构中并被组织为分层数据树;

用于接收统一资源定位器的装置,从客户端计算机接收所述统一资源定位器,其中,所述统一资源定位器至少包括指定所述数据树的第一视图的部分;

用于创建关系数据库查询的装置,利用所述统一资源定位器的所述部分来创建所述关系数据库查询,并将所述查询提交给关系数据库;

用于接收数据元素集的装置,接收来自所述关系数据库的与所述第一视图相关联的所述数据元素集;

用于生成第一电子文档的装置,包括对应于对所述数据树中的节点执行动作而生成通向其他统一资源定位器的变形链接,所述第一电子文档包含所述数据元素集;

用于将所述第一电子文档提供给所述客户端计算机的装置。

2. 根据权利要求 1 所述的设备,其中,所述用于接收统一资源定位器的装置包括用于接收这样的统一资源定位器的装置,该统一资源定位器包含所述数据树的打开节点的一个或多个标识符的列表。

3. 根据权利要求 2 所述的设备,其中,所述用于创建关系数据库查询的装置包括用于创建结构化查询语言关系数据库查询的装置,在该结构化查询语言关系数据库查询中,打开节点的所述标识符中的至少一个是 SELECT 语句的参数。

4. 根据权利要求 3 所述的设备,其中,所述用于创建结构化查询语言关系数据库查询的装置包括用于创建这样的结构化查询语言关系数据库查询的装置,该结构化查询语言关系数据库查询指定只选择其分层双亲全部处于打开节点的列表中的那些数据元素的结果集。

5. 根据权利要求 2 所述的设备,其中,所述用于接收统一资源定位器的装置包括用于接收包含所述第一视图的视图类型的所述统一资源定位器的装置。

6. 根据权利要求 5 所述的设备,其中,所述用于生成第一电子文档的装置包括:用于生成包含所述数据元素集的第一电子文档的装置,包括用于对应于打开或关闭所述数据树中的节点而生成通向其他统一资源定位器的变形链接的装置。

7. 根据权利要求 1 所述的设备,其中,所述用于生成第一电子文档的装置包括:用于生成包含所述数据元素集的第一电子文档的装置,包括用于对应于打开或关闭所述数据树中的节点而生成通向其他统一资源定位器的变形链接的装置;其中,对应于打开节点的第一变形链接包括:所接收到的向标识符的所述列表添加了当前关闭节点的标识符的统一资源定位器;其中,对应于关闭节点的第二变形链接包括:所接收到的从标识符的所述列表移除了当前打开节点的标识符的统一资源定位器。

8. 一种用于传送和显示分层数据的机器实现方法,包括:

接收来自客户端计算机的统一资源定位器,其中,所述统一资源定位器至少包括指定分层数据树的第一视图的部分,所述分层数据树表示储存在数据储存器中的包含在关系数据库结构中的多个数据元素;

利用所述统一资源定位器的所述部分来创建关系数据库查询并将所述查询提交给关

系数据库；

接收来自所述关系数据库的与所述第一视图相关联的数据元素集；

生成包含所述数据元素集的第一电子文档，包括对应于对所述数据树中的节点执行动作而生成通向其他统一资源定位器的变形链接；

将所述第一电子文档提供给所述客户端计算机。

9. 根据权利要求 8 所述的方法，其中，接收所述统一资源定位器包括：接收包含所述数据树的打开节点的一个或多个标识符的列表的所述统一资源定位器。

10. 根据权利要求 9 所述的方法，其中，创建所述关系查询包括：创建结构化查询语言关系数据库查询，在该结构化查询语言关系数据库查询中，打开节点的所述标识符中的至少一个是 SELECT 语句的参数。

11. 根据权利要求 10 所述的方法，其中，创建所述结构化查询语言关系数据库查询包括创建这样的结构化查询语言关系数据库查询，该结构化查询语言关系数据库查询指定只选择其分层双亲全部处于打开节点的列表中的那些数据元素的结果集。

12. 根据权利要求 8 所述的方法，其中，生成所述第一电子文档包括：生成包含所述数据元素集的第一电子文档，包括对应于打开或关闭所述数据树中的节点而生成通向其他统一资源定位器的变形链接；其中，对应于打开节点的第一变形链接包括：所接收到的向标识符的所述列表添加了当前关闭节点的标识符的统一资源定位器；其中，对应于关闭节点的第二变形链接包括：所接收到的从标识符的所述列表移除了当前打开节点的标识符的统一资源定位器。

13. 一种用于传送和显示分层数据的机器，包括：

第一装置，用于接收来自客户端计算机的统一资源定位器，其中，所述统一资源定位器至少包括指定分层数据树的第一视图的部分，所述分层数据树表示储存在数据储存器中的包含在关系数据库结构中的多个数据元素；

第二装置，用于利用所述统一资源定位器的所述部分来创建关系数据库查询并将所述查询提交给关系数据库；

第三装置，用于接收来自所述关系数据库的与所述第一视图相关联的数据元素集；

第四装置，用于生成包含所述数据元素集的第一电子文档，包括对应于对所述数据树中的节点执行动作而生成通向其他统一资源定位器的变形链接；

第五装置，用于将所述第一电子文档提供给所述客户端计算机。

14. 根据权利要求 13 所述的机器，其中，用于接收所述统一资源定位器的第一装置包括：用于接收包含所述数据树的打开节点的一个或多个标识符的列表的所述统一资源定位器的装置。

15. 根据权利要求 14 所述的机器，其中，用于创建所述关系查询的第二装置包括用于创建结构化查询语言关系数据库查询的装置，在该结构化查询语言关系数据库查询中，打开节点的所述标识符中的至少一个是 SELECT 语句的参数。

16. 根据权利要求 15 所述的机器，其中，用于创建所述结构化查询语言关系数据库查询的装置包括用于创建这样的结构化查询语言关系数据库查询的装置，该结构化查询语言关系数据库查询指定只选择其分层双亲全部处于打开节点的列表中的那些数据元素的结果集。

17. 根据权利要求 13 所述的机器,其中,用于生成所述第一电子文档的第四装置包括用于生成包含所述数据元素集的第一电子文档,对应于打开或关闭所述数据树中的节点而生成通向其他统一资源定位器的变形链接的装置;其中,对应于打开节点的第一变形链接包括:所接收到的向标识符的所述列表添加了当前关闭节点的标识符的统一资源定位器;其中,对应于关闭节点的第二变形链接包括:所接收到的从标识符的所述列表移除了当前打开节点的标识符的统一资源定位器。

用于传送和显示分层数据的设备以及机器实现方法

技术领域

[0001] 本公开文件大体涉及在电子文档中（诸如互联网网页上）显示来自（诸如大型关系数据库）数据库的数据。

[0002] 背景技术

[0003] 可以执行在这部分中所描述的方法，但是没有必要是之前已经构想出的或执行过的方法。因此，除非在本文中另有说明，否则在这部分中所描述的方法不是相对于本申请中的权利要求的现有技术，并且不认为这些方法是包含在这部分中的现有技术。

[0004] 当今的互联网提供了用于向世界范围的计算机分配数字化信息的极其有效的手段，而网络浏览器界面则成为用于显示该信息的一种极好的手段。

[0005] 该分配 / 显示模型的一种普遍的实现方式是基于从主机上的数据库提取的特定信息而使用动态生成的网页，在该网页中，用户指定的统一资源定位器（URL）触发网页的即时创建。

[0006] 尽管必须使用网络应用语言（web application language）（诸如 PHP 或 Cold Fusion）之一来编写网页的动态结构，基于用户的 URL 派生的（URL-derived）选择参数并且还基于主机数据库中可能变化的值，单个这样的动态网页仍可以提供无限多的不同数据显示。由于只需一次且对一个页面编程，所以该模型的效率极高，其已经成为用于大多数网站的标准方法。

[0007] 利用最常见的引擎中的 MySQL、Oracle、及 SQL-Server 引擎，通常基于 SQL 数据库语言，数据本身可能越来越多地位于关系数据库中。这种关系数据库提供了极高的性能、储存效率、及潜在大小的巨大可扩展性。

[0008] 尽管该数据分配 / 显示模型具有这些显著的优点，其在方便地应用于各种类型的数据方面仍面临着一些障碍。部分地由于这个原因，互联网数据很少像位于一个人自己的计算机上的软件应用程序中的数据那样快速且容易地显示和操作。

[0009] 一个障碍在于用来确定在网页上示出的特定数据的 URL 在结构上具有显著的限制。变得过于长或过于复杂的 URL 对于个人来说不便于理解、操作、或存储为链接。对所嵌入的短的和不太复杂的 URL 的信息内容进行最大化极大地提高了网络显示系统的效率。

[0010] 另一障碍在于在互联网上潜在可获得的许多数据集尺寸巨大并且复杂。将单个动态网页编写为便于对这种复杂的数据集进行显示或操纵，可能是一项困难的任務，并且容易出错。

[0011] 最后，无论是由于目前用于在主机数据库和网页之间选择和传送数据的多种方法的下层数据库算法，还是由于这些方法的显示技术，这些方法均看起来缓慢且低效。因此，主机数据库会变得过载，客户机或服务器带宽变得堵塞，而整体性能经常会很缓慢。

[0012] 总之，这些困难或类似困难极大地限制了有效显示或操纵基于互联网的数据。

[0013] 许多这些预期的较大的互联网数据集在结构上是分层的，其被组织为收集成的一族不同普遍性等级的数据元素。这种分层数据集的最普遍的实例可能就是特定计算机上的目录和文件，其中，“孩子”目录可能包含在另外的“双亲”目录中等等，一直到计算机存储器

的最高等级。

[0014] 使分层数据直观化或对分层数据进行显示的标准手段是按照“树”结构的形式,诸如计算机的硬盘的目录/文件树。这方便地使得用户能够通过“打开”或“关闭”树的不同节点或分支而按照所期望的任意详细程度来总结或检查数据。

[0015] 大多数的大型分层数据集或者当前存储在了关系数据库中或者能够迅速和容易地载入到这些数据库表格中。关系数据库对其所包含的数据集设置了某些重要的限制,但也从而为以后操纵或选择该数据提供了极高的效率。

[0016] 作为实例,尽管计算机硬盘驱动器的文件和目录的描述信息通常不存储在关系数据库表格中,仍可以容易地将描述信息载入到这样的表格中,利用作为可能从全路径名派生出的唯一标识符的一个字段、作为元素类型(文件或目录)的另一字段、作为双亲目录的标识符的另一字段、以及作为与单独的文件或目录相关联的各种其他类型的附加信息的其他字段。

[0017] SQL 数据库是最普遍的关系数据库实例,并且通常要求为数据元素分配一个或多个唯一标识符,其中,由数据库引擎本身自动存储和索引这些唯一标识符用于快速检索。在其他特性中,SQL 数据库引擎最适合于检索由唯一或非唯一索引值所选定的数据,也最适合于基于这些索引值而将数据库表格与其本身或者与其他表格合并或“结合”在一起。

[0018] 可以写成一条单独的、甚至是高度复杂的 SQL 语句(可能包括多个联结或自联结)的数据库查询常常比不太高效的数据库系统中的一组等价的数据库操作快几个数量级。

附图说明

[0019] 在附图的示图中以实例的方式而不是以限制的方式来阐述本发明,其中,相同的参考标号表示相同的元件,在附图中:

[0020] 图 1 示出了表示示例性实施例的操作顺序的流程图。

[0021] 图 2 示出了其中目录 REPORT 打开的示例性显示视图。

[0022] 图 3 示出了不同的显示视图。

[0023] 图 4 提供了能够从选择器 URL 产生 HTML 目录/文件树结构的 SQL SELECT 语句的实例。

[0024] 图 5 提供了表示示例性实施例的主要部分的框图。

[0025] 图 6 示出了在其上可以实现实施例的计算机系统。

具体实施方式

[0026] 在下面的描述中,为了进行阐述,陈述了许多具体细节以提供对本发明的透彻理解。然而,对于本领域技术人员来说显而易见的是,不用这些具体细节也可以实施本发明。在其他实例中,按照框图的形式示出公知的结构和装置,以避免不必要地模糊本发明。

[0027] 对用于传送和显示数据库和电子文档之间的分层数据的方法加以描述。在一个实施例中,提供了一种新的软件方法,用于传送和显示关系数据库和互联网网页之间的分层数据,旨在提高这些处理的速度、效率、以及简单性。在实施例中,按照数据树的形式来显示所选择的数据,其具体的显示视图由网络位置标识符(networklocation identifier)(诸如统一资源定位器(URL))的一部分来确定,其中,该字符串的同一部分还被翻译为用于从

数据库中选择数据的查询命令的一部分。

[0028] 在一个实施例中,数据处理设备包括:数据储存器,连接至一个或多个处理器,以及包含在关系数据库结构中并被组织为分层数据树的多个数据元素,以及执行接收来自客户端计算机的网络位置标识符的指令,其中,网络位置标识符至少包括指定数据树的第一视图的部分;利用网络位置标识符的该部分来创建关系数据库查询,并将该查询提交给关系数据库;接收来自关系数据库的与第一视图相关联的数据元素集;生成包括该数据元素集的电子文档,包括对应于在数据树中的节点上执行动作而生成通向其他网络位置标识符的变形链接(transformative link);以及将的一电子文档提供给客户端计算机。

[0029] 在实施例中,接收网络位置标识符包括:接收包括数据树的打开节点的一个或多个标识符的列表的网络位置标识符。在实施例中,创建关系数据库查询包括:创建结构化查询语言(SQL)查询,其中,打开节点的标识符中的至少一个标识符是SELECT语句的参数。在实施例中,创建SQL查询包括创建这样的SQL查询,该SQL查询指定只选择其分层双亲全部处于打开节点的列表中的那些数据元素的结果集。

[0030] 在实施例中,生成第一电子文档包括:生成包括该数据元素集的电子文档,包括对应于打开或关闭数据树中的节点而生成通向其他网络位置标识符的变形链接;其中,对应于打开节点的第一变形链接包括:所接收到的向标识符的列表添加了当前关闭节点的标识符的网络位置标识符;其中,对应于关闭节点的第二变形链接包括:所接收到的从标识符的列表移除了当前打开节点的标识符的网络位置标识符。

[0031] 其他方面和特征将从以下的说明书、附图、及所附权利要求中而变得显而易见。

[0032] 在本发明的示例性实施例的情况下,由打开树节点的列表来确定数据树的显示形式,该打开树节点的列表包含在网络位置标识符的特定字段中,其可以是URL字符串。

[0033] 提取出网络位置标识符的该字段,然后将该字段嵌入到数据库查询中,其产生被限制到分层双亲全部处于打开节点列表中的那些数据元素的结果集。完成选择之后,用网络应用语言对这些数据元素进行处理,并按照所期望的格式将这些数据元素连同任意的可选链接或按钮动作一起显示在动态网页上并示出所期望的信息。

[0034] 在实施例中,关系数据库是MySQL,网络应用语言是ColdFusion,网页服务器是Apache,数据库查询表示为SQL SELECT语句,而数据元素本身包括与数据存储系统的一部分的特定文件和目录有关的信息。

[0035] 如果将要呈现的分层数据还不是关系数据库格式的,则首先将分层数据加载为这样的格式。

[0036] 在实施例中,所示出的数据树的具体视图完全由其打开节点列表决定。因此,网络位置标识符选择器的打开节点部分完全决定了数据树的视图。在关系数据库中可以用这些节点的唯一标识符来表示这些节点。

[0037] 尽管由用户来决定所显示的数据树的确切实施方式,但一个实例是完全按照HTML显示的文本树。

[0038] 所显示的文本数据树中的每个节点还可以包括其网络位置标识符控制的动作会是关闭所打开的节点或打开所关闭的节点的链接。可以将特定的系统设计为要求特定的树节点一直打开或一直关闭。

[0039] 关系数据库查询使用URL的相关部分的确切形式并不重要,并且可以根据特定的

数据库语言、网页应用语言、以及系统设计而有所不同。

[0040] 图 1 示出了表示本发明示例性实施例的操作顺序的流程图。首先,客户端机器(client machine)上的用户按照用于网站的特定 URL 的形式来指定网络位置标识符[#10],该 URL 由服务器机器上的网站系统接收[#11]。接下来,网站系统对该 URL 系统进行解析,并提取决定所显示的数据树视图的那部分[#12]。然后,将该 URL 字段插入到数据树 SQL 数据库查询中[#13],其返回来自 SQL 数据库引擎的作为结果的显示项目集[#14]。然后,网站系统按照数据树的形式显示这些返回的元素,将适当的链接插入到表示不同的数据树视图的新 URL 中,并发送作为结果的 HTML 页面用于在用户的机器上显示[#15]。然后,用户可以通过选择 HTML 页面上的各种 URL 链接来在该页面上选择打开或关闭任意的数据节点,以表示数据树的不同显示视图[#16]。

[0041] 在图 2 和图 3 中,命名为 REPORT 的特定文件目录及其各种子目录和文件被作为简单的 HTML 文本数据树显示、在两个不同的显示视图中示出、并由两个不同的 URL 字段决定。

[0042] 在图 2 中,显示视图为目录 REPORT[#10] 打开,子目录 CG[#11] 和子-子目录 CGS[#12] 亦然。该视图由 URL 的 OpenNodes 部分决定[#13],该 URL 包含字段“OpenNodes = 96278,96281,96690”(那三个打开的目录的唯一 ID 代码的列表)。

[0043] 表示目录的文本树的每一行均包含初始文本符号链接[X](在该目录关闭的情况下)和[-](在该目录打开的情况下)。对这些链接进行选择会在打开和关闭之间自动切换给定目录,即打开所关闭的目录以及关闭所打开的目录。

[0044] 打开的链接包含当前的 URL,但是具有附加到 OpenNodes 字段的当前关闭的目录的 ID。关闭的链接包含当前的 URL,但是从 OpenNodes 字段移除了当前打开的目录的 ID。例如,图 2 中临近目录 2002 的打开链接符号[X]包含 URL 字段“OpenNodes = 96278,96281,96690,96694”,其中,该列表中的最后一个 ID 对应于目录 2002。因此,点击该链接将重新加载同一网页,但是,其中,表示目录 2002 的新 URL 会被示为打开而不是关闭。这个新 URL 对应于图 3 中所示的不同显示视图。

[0045] 图 4 提供了能够从选择器 URL 产生 HTML 目录/文件树结构的 SQL SELECT 语句的实例。SQL SELECT 语句嵌入在 ColdFusion 网络应用语言中所表达的函数调用中。示例性的 ColdFusion/SQL 语句返回将要在图 2 中的屏幕上显示的那些文件或目录(即,其双亲包含在打开节点列表中的所有那些文件或目录)。

[0046] 在 SQL 语句中,变量 ID 表示特定文件或目录的唯一标识符,parent ID 是双亲目录的唯一标识符,而 Level 表示文件或目录的等级,其中,最高等级目录为 0,其子目录为 1,其子-子目录为 2,等等。所提供的特定示例性语句能够显示 4 个等级深度的目录结构。

[0047] 在所描述的 SQL 查询已经返回了将要在屏幕上显示的文件和目录之后,标准的网络应用编程可以用于生成 HTML 树的显示格式。

[0048] 在正常环境下,只需要如本文所描述的单个 SQL SELECT 语句来生成用于任意指定大小的分层数据树的任意显示视图的值。因而,尽量少对数据库服务器施加压力,从而响应极快。

[0049] 此外,这些方法通过提供选择 URL 和数据库选择查询之间的自然映射而使得设计动态网页所需的编程的复杂性最小化。另外,由于通常给定数据树的相对少的节点在给定时间打开,因此该方法使得选择 URL 的大小和复杂性最小化。

[0050] 图 5 提供了表示示例性实施例的主要部分的框图。客户端机器 [#10] 直接地或通过一个或多个网络间接地连接至主管 (host) 网络服务器、应用服务器、以及数据库服务器的一个或多个计算机。在一个实施例中,网络服务器是 Apache 网络服务器 [#11],应用服务器是 Cold Fusion 网络应用服务器 [#12],数据库服务器是 MySQL 关系数据库服务器 [#13],MySQL 关系数据库服务器主管一个或多个 MySQL 关系数据库表格 [#14]。

[0051] 图 6 是示出了在其上可以实施本发明实施例的计算机系统 600 的框图。计算机系统 600 包括总线 602 或用于交流信息或其他通信机构、以及与总线 602 连接的用于处理信息的处理器 604。计算机系统 600 还包括主存储器 606,诸如随机存取存储器 (“RAM”) 或其他动态储存装置,主存储器连接至总线 602,用于存储由处理器 604 执行的信息和指令。在由处理器 604 执行将要执行的指令期间,主存储器 606 还可以用于存储临时变量或其他中间信息。计算机系统 600 进一步包括只读存储器 (“ROM”) 608 (或其他静态储存装置),该只读存储器连接至总线 602,用于存储用于处理器 604 的静态信息和指令。提供了储存装置 610 (诸如磁盘或光盘),该储存装置连接至总线 602,用于存储信息和指令。

[0052] 计算机系统 600 可以经由总线 602 连接至用于向计算机用户显示信息的显示器 612,诸如阴极射线管 (“CRT”)。输入装置 614 (包括字母数字和其他按键) 连接至总线 602,用于向处理器 604 传送信息和命令选择。另一类型的用户输入装置是光标控制器 616,诸如鼠标、跟踪球、指示笔、或光标方向键,用于向处理器 604 传送方向信息和命令选择以及用于控制显示器 612 上的光标移动。该输入装置通常在两个轴 (第一轴 (例如, x) 和第二轴 (例如, y)) 上具有两个自由度,这允许该装置在平面中指定位置。

[0053] 本发明涉及使用计算机系统 600 来传送并显示数据库与电子文档之间的分层数据。根据本发明的一个实施例,响应于处理器 604 执行包含在主存储器 606 中的一个或多个指令的一个或多个序列而由计算机系统 600 提供数据库和电子文档之间的分层数据的传送和显示。可以将这些指令从另一计算机可读介质 (诸如储存装置 610) 中读取到主存储器 606 中。执行包含在主存储器 606 中的指令序列使得处理器 604 执行本文所描述的处理步骤。在可替换的实施例中,可以使用硬线电路替换软件指令或者与软件指令结合来实施本发明。因此,本发明的实施例不限于硬件电路与软件的任何特定组合。

[0054] 本文所使用的术语“计算机可读介质”指参与向用于执行的处理器 604 提供指令的任意介质。这样的介质可以采用多种形式,包括但不限于非易失性介质、易失性介质、以及传输介质。非易失性介质包括 (例如) 光盘或磁盘 (诸如储存装置 610)。易失性介质包括动态存储器 (诸如主存储器 606)。传输介质包括同轴电缆、铜线、及光纤,包括包含总线 602 的导线。传输介质还可以采用声波或光波的形式,诸如在无线电波和红外数据通信期间所产生的那些声波或光波。

[0055] 计算机可读介质的普遍形式包括 (例如) 软盘、软磁盘、硬盘、磁带、或任意其他磁性介质、CD-ROM、任意其他光学介质、穿孔卡片、纸带、具有孔图案的任意其他物理介质、RAM、PROM、及 EPROM、FLASH-EPROM、任意其他存储芯片或盒式磁盘、下文中所描述的载波、或计算机可读的任意其他介质。

[0056] 计算机可读介质的各种形式可以涉及携带一个或多个指令的一个或多个序列至用于执行的处理器 604。例如,这些指令可以最初携带在远程计算机的磁盘上。远程计算机可以将这些指令加载到其动态存储器,并使用调制解调器通过电话线来发送这些指令。计

计算机系统 600 本地的调制解调器可以接收电话线上的数据并利用红外发射机将数据转换为红外信号。红外检测器可以接收红外信号中携带的数据,适当的电路可以将该数据置于总线 602 上。总线 602 将该数据携带到主存储器 606,处理器 604 从主存储器中检索并执行这些指令。在这些指令由处理器 604 执行前或执行后,主存储器 606 所接收到的指令可以优选地存储在储存装置 610 上。

[0057] 计算机系统 600 还包括连接到总线 602 的通信接口 618。通信接口 618 提供连接到网络链路 620 的双向数据通信,网络链路连接到局部网络 622。例如,通信接口 618 可以是综合服务数字网(“ISDN”)卡或调制解调器,用来向相应类型的电话线提供数据通信连接。作为另一实例,通信接口 618 可以是局域网(“LAN”)卡,用来向兼容 LAN 提供数据通信连接。还可以实现无线链路。在任意这样的实施方案中,通信接口 618 发送和接收电、电磁、或光信号,这些信号携带表示各种类型的信息的数字数据流。

[0058] 网络链路 620 通常通过一个或多个网络向其他数据装置提供数据通信。例如,网络链路 620 可以通过局部网络 622 向主机 624 或者向由互联网服务提供商(“ISP”)626 所操作的数据设备提供连接。而 ISP 626 通过全世界的数据包数据通信网络(现在普遍称作“互联网”628)来提供数据通信服务。局部网络 622 和互联网 628 均使用携带数字数据流的电、电磁、或光信号。通过各种网络的信号以及网络链路 620 上和通过通信接口 618 的信号(其携带去往以及来自计算机系统 600 的数字数据)是传送信息的载波的示例性形式。

[0059] 计算机系统 600 可以通过网络(多个网络)、网络链路 620、及通信接口 618 来发送消息和接收数据(包括程序代码)。在互联网实例中,服务器 630 可能通过互联网 628、ISP 626、局部网络 622、及通信接口 618 来传送所请求的应用程序代码。根据本发明,所下载的这样的应用程序提供如本文所描述的数据库和电子文档之间的分层数据的传送和显示。

[0060] 在接收到代码时,可以由处理器 604 来执行所接收到的代码,和/或将代码存储在储存装置 610 或其他非易失性存储器中用于以后执行。如此,计算机系统 600 可以获得载波形式的应用程序代码。

[0061] 在多种可替换实施例中的一个实施例中,显示数据树的动态网页还可以按照 XML 或现有的或未来的某种其他网页格式呈现,而不是由 HTML 文本组成。

[0062] 在另一可替换实施例中,用于生成动态网页的模版可以改为使用现有的或未来的某种其他网络应用编程语言,诸如 PHP,而非基于 ColdFusion。可以使用现有的或未来的某种其他网页服务器,而非 Apache。驱动这些动态网页的创建的下层数据库系统可以改为依照 Oracle SQL、Microsoft SQL-Server、或者现有的或未来的某种其他 SQL 或其他关系数据库。

[0063] URL 选择器的相关部分可以使用某种其他方法来对数据树的显示视图进行编码,而不是表示所显示的数据树的打开节点。此外,可以通过串联多个独立的数据树的 URL 选择器或另外使用其他方法来对它们进行合并来同时显示和控制多个独立的数据树。

[0064] 在上述的说明书中,通过参考本发明的特定实施例描述了本发明。然而,显而易见的是,在不背离本发明的较宽的精神和范围的情况下,可以对其进行各种修改和改变。因此,应当将说明书和附图视为示例性的而不是限制意义的。

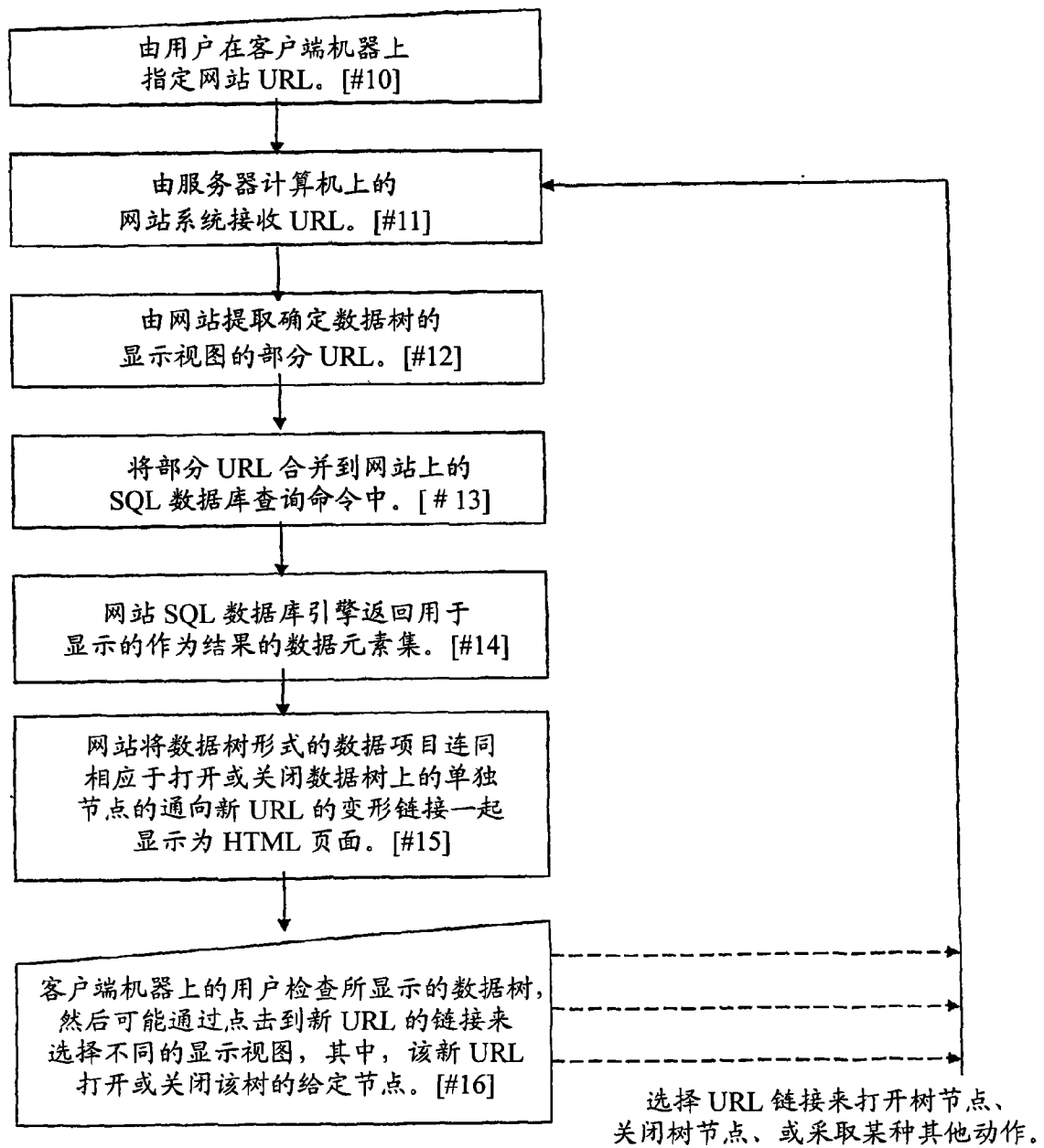


图 1

[Login](#) [HOME](#) | [Authors](#) | [Publications](#) | [Organizations](#) | [Libraries](#) | [Users](#) | [ID=4202]
[TESTS](#) | [Inventory](#) | [System](#) | [Admin]

System Administration [Currently \[Admin\]](#) [Toggle User Mode](#)
[\[Processing\]](#) | [\[System\]](#) | [\[User Accounts\]](#) | [\[User Activity\]](#) | [\[Hit Activity\]](#) | [\[Analysis\]](#) | [\[Help Editor\]](#)
[\[Files\]](#)

URL=[View=Basic&OpenNodes=96278,96281,96690]
Sort by: [\[Type\]](#) | [\[Size\]](#) | [\[Name\]](#) | [\[Extension\]](#) | [\[Date\]](#) / [\[Ascending\]](#) | [\[Descending\]](#)

- [\[-\]](#) **REPORT [96278]**
 - 1. **AN** = 1 Dir, 6 MB [96279]
 - 2. **CE** = 1 Dir, 290 MB [96280]
 - 3. **CG** [96281]
 - 1. **CGS** [96690]
 - 1. **1998** = 5 Files, 343 KB [96691]
 - 2. **2000** = 5 Files, 451 KB [96692]
 - 3. **2001** = 6 Files, 617 KB [96693]
 - 4. **2002** = 15 Files, 4 MB [96694]
 - 5. **2003** = 21 Files, 12 MB [96695]
 - 6. **2004** = 10 Files, 13 MB [96696]
 - 4. **CI** = 1 Dir, 158 MB [96282]
 - 5. **FR** = 1 Dir, 6 MB [96283]
 - 6. **LE** = 1 Dir, 33 MB [96284]
 - 7. **NC** = 1 Dir, 15 MB [96285]
 - 8. **NI** = 1 Dir, 101 MB [96286]
 - 9. **PP** = 1 Dir, 103 MB [96287]
 - 10. **PU** = 1 Dir, 29 MB [96288]
 - 11. **RE** = 1 Dir, 49 MB [96289]
 - 12. **SO** = 1 Dir, 4 MB [96290]
 - 13. **TR** = 1 Dir, 12 MB [96291]

图 2

[Login](#) | [HOME](#) | [Authors](#) | [Publications](#) | [Organizations](#) | [Libraries](#) | [Users](#) | [ID=4202]
[TESTS](#) | [Inventory](#) | [System](#) | [\[Admin\]](#)

System Administration Currently [\[Admin\]](#) [Toggle User Mode](#)
[\[Processing\]](#) | [\[System\]](#) | [\[User Accounts\]](#) | [\[User Activity\]](#) | [\[Hit Activity\]](#) | [\[Analysis\]](#) | [\[Help Editor\]](#)
[\[Files\]](#)

URL=[\[View=Basic&OpenNodes=96278,96281,96690,96694\]](#)
 Sort by: [\[Type\]](#) | [\[Size\]](#) | [\[Name\]](#) | [\[Extension\]](#) | [\[Date\]](#) / [\[Ascending\]](#) | [\[Descending\]](#)

- [\[-\]](#) **REPORT [96278]**
 - 1. **AN = 1 Dir, 6 MB [96279]**
 - 2. **CE = 1 Dir, 290 MB [96280]**
 - 3. [\[-\]](#) **CG [96281]**
 - 1. [\[-\]](#) **CGS [96690]**
 - 1. **1998 = 5 Files, 343 KB [96691]**
 - 2. **2000 = 5 Files, 451 KB [96692]**
 - 3. **2001 = 6 Files, 617 KB [96693]**
 - 4. [\[-\]](#) **2002 [96694]**
 - 1. **CGS_Report-2002-Cover.jpg = 812 KB [96713]**
 - 2. **CGS_Report-2002-Cover5mini.jpg = 4 KB [96714]**
 - 3. **CGS_Report-2002-Cover5thumb.jpg = 7 KB [96715]**
 - 4. **CGS_Report-2002-Cover5tiny.jpg = 856 Bytes [96716]**
 - 5. **CGS_Report-2002.pdf = 3 MB [96717]**
 - 6. **CGS_RyanPaul-2002-Cover.jpg = 156 KB [96718]**
 - 7. **CGS_RyanPaul-2002-Cover5mini.jpg = 2 KB [96719]**
 - 8. **CGS_RyanPaul-2002-Cover5thumb.jpg = 3 KB [96720]**
 - 9. **CGS_RyanPaul-2002-Cover5tiny.jpg = 563 Bytes [96721]**
 - 10. **CGS_RyanPaul-2002.pdf = 225 KB [96722]**
 - 11. **CGS_RyanPaul-2002n02-Cover.jpg = 153 KB [96723]**
 - 12. **CGS_RyanPaul-2002n02-Cover5mini.jpg = 2 KB [96724]**
 - 13. **CGS_RyanPaul-2002n02-Cover5thumb.jpg = 3 KB [96725]**
 - 14. **CGS_RyanPaul-2002n02-Cover5tiny.jpg = 543 Bytes [96726]**
 - 15. **CGS_RyanPaul-2002n02.pdf = 222 KB [96727]**
 - 5. **2003 = 21 Files, 12 MB [96695]**
 - 6. **2004 = 10 Files, 13 MB [96696]**
 - 4. **CI = 1 Dir, 158 MB [96282]**
 - 5. **FR = 1 Dir, 6 MB [96283]**
 - 6. **LE = 1 Dir, 33 MB [96284]**
 - 7. **NC = 1 Dir, 15 MB [96285]**
 - 8. **NI = 1 Dir, 101 MB [96286]**
 - 9. **PP = 1 Dir, 103 MB [96287]**
 - 10. **PU = 1 Dir, 29 MB [96288]**
 - 11. **RE = 1 Dir, 49 MB [96289]**
 - 12. **SO = 1 Dir, 4 MB [96290]**
 - 13. **TR = 1 Dir, 12 MB [96291]**

图 3

```
[URL_string is "View=Basic&OpenNodes=96278,96281,96690"]  
  
<cfset OpenNodeList =  
  REReplace(URL_string, ".*OpenNodes=([0-9,]+).*", "\1")>  
  
<cfquery name="DirectoryTree" datasource="FilesDB">  
  SELECT L0.* FROM files AS S,  
    files AS L3,files AS L2,files AS L1,files AS L0  
  WHERE S.ID=96278 AND (L0.ID=S.ID OR L0.Level > S.Level)  
  AND L3.ID=L2.parentID AND L2.ID=L1.parentID AND L1.ID=L0.parentID  
  AND L0.parentID IN (#OpenNodeList#)  
  AND L1.parentID IN (#OpenNodeList#)  
  AND L2.parentID IN (#OpenNodeList#)  
  AND L3.parentID IN (#OpenNodeList#)  
  ORDER BY CASE L0.Level  
    WHEN 0 THEN L0.ID  
    WHEN 1 THEN CONCAT_WS('-', L1.ID, L0.ID)  
    WHEN 2 THEN CONCAT_WS('-', L2.ID, L1.ID, L0.ID)  
    WHEN 3 THEN CONCAT_WS('-', L3.ID, L2.ID, L1.ID, L0.ID)  
    ELSE L0.ID END</cfquery>
```

图 4

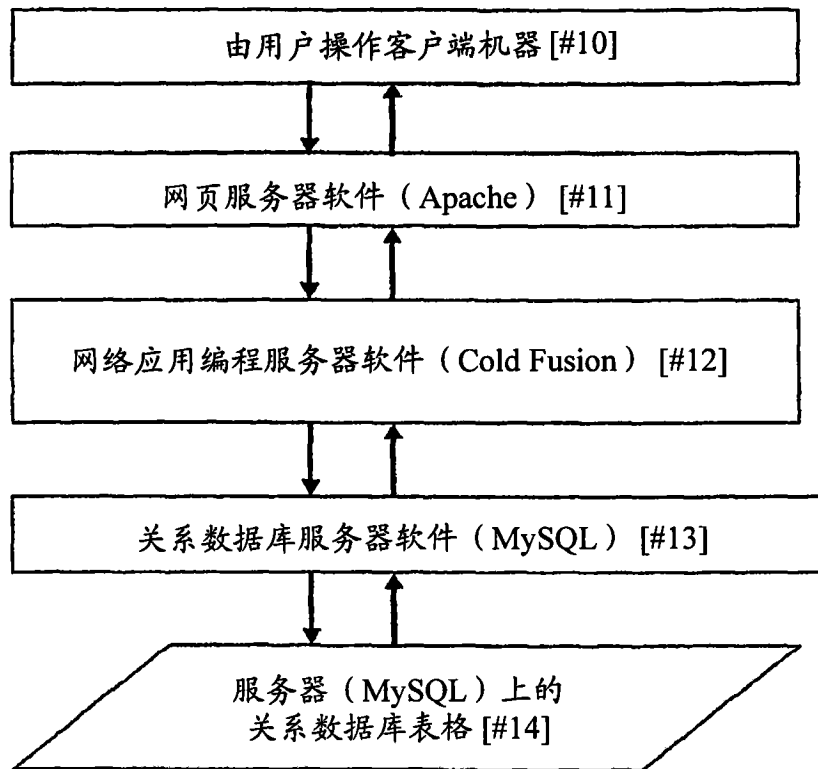


图 5

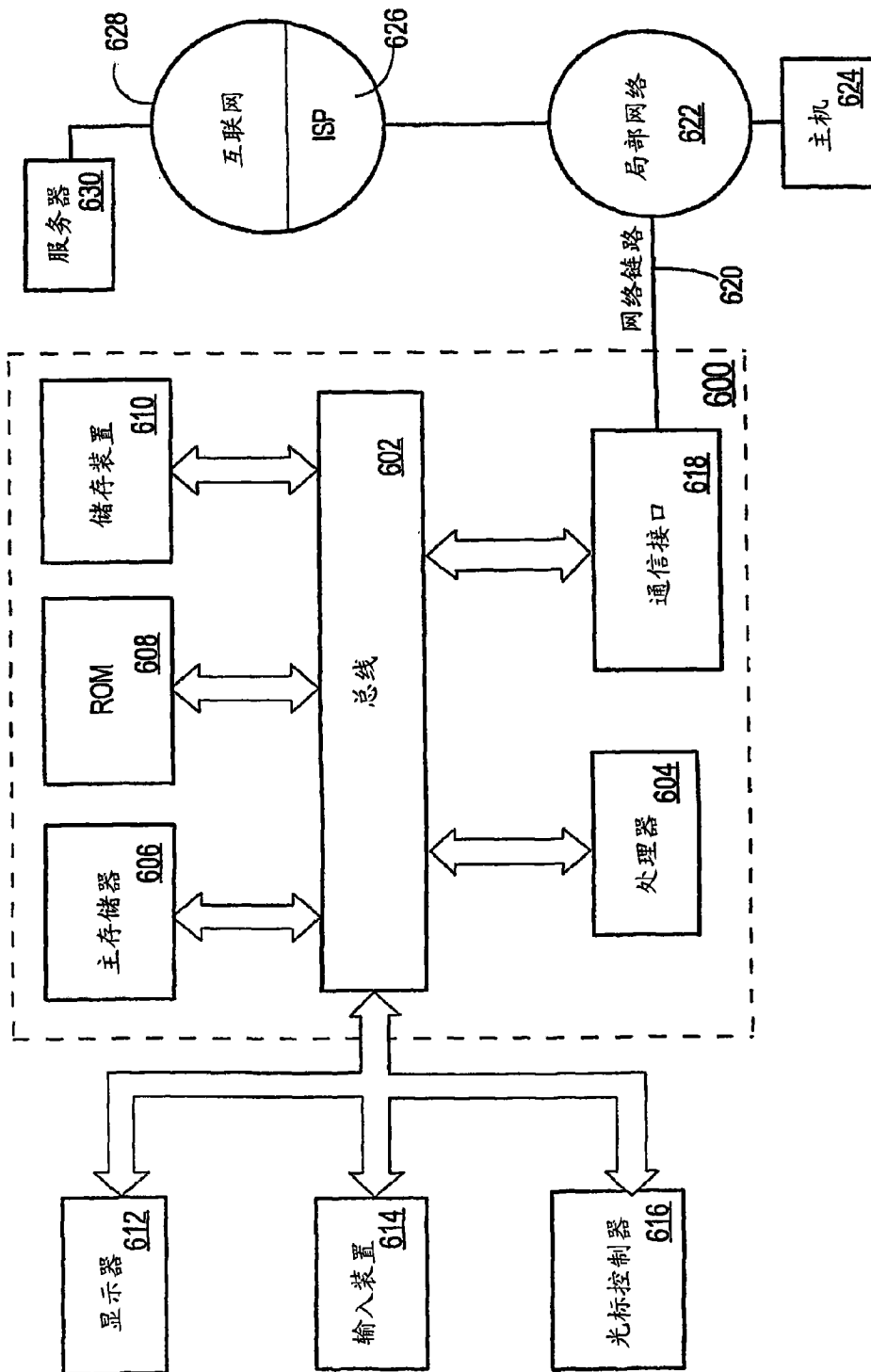


图 6