

财务分析器

OLAP

联机分析处理

报表

(15)

通用财务 OLAP 分析器的实现<sup>\*</sup>

Implementation of General Financial OLAP Analytical System

62-65/40

刘瑶 孙玉芳

F234.4

(中国科学院软件研究所 北京 100080)

**Abstract** On Line Analytical Processing (OLAP) is one of the newest and general technologies of late years. In this paper, we present the introduction of OLAP into accounting analysis area and implementation of the financial analysis system.

Firstly, the paper presents the system architecture and the module's function. Secondly, the paper tells about the design system implementation. Thirdly, the paper lays emphasis on the analytical system implementation. Based on analysis, there are some restrictions for straightly using the OLAP model to realize financial analysis. The paper expands the OLAP model, implements the operation of variable granularity and analysis tree, and makes advances in the navigation charts.

**Keywords** OLAP, Analytical model, Variable granularity, Analysis tree, Design tool, Analytical system

## 1. 引言

仅仅使用报表作为财务分析的手段最易实现,但分析者使用起来不方便,因为分析者需要的数据分散于各个报表之中,没有系统、合理而有效地组织起来。

如果将 OLAP<sup>[1]</sup> 技术应用于财务分析,不仅可以利用 OLAP 的钻取、切片、漂移、旋转等功能简单而直观地引导用户从多角度、多侧面观察数据,而且通过对 OLAP 功能的增强,可以将分析数据统一管理,使用户在同一界面下选择分析的重点,获取所有分析数据。

本文讲述通用财务 OLAP 分析器的实现,给出其系统结构,描述主要模块功能,阐述设计器实现和分析器实现,特别是灵活粒度和分析树的处理,并对导航图进行了改进。

## 2. 系统结构

图 1 是整个财务分析器系统的框架,其中主要部分是 OLAP 分析器。

一个完整的 OLAP 系统不仅要能够提供 OLAP 的各种分析手段,而且还要提供包括 OLAP 系统的初始准备、设置和连续操作等管理功能。本系统划分为两大独立模块:设计器和分析器。设计器完成系统的初始准备、权限设置、模型创建等功能,而分析器则完成与

OLAP 相关的分析操作。

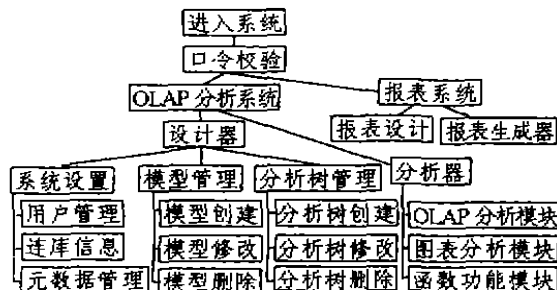


图 1 通用财务分析器系统结构框架

## 3. 主要模块功能

图中标注的报表系统实现详见文[2],这里简要介绍 OLAP 分析系统中设计器和分析器的主要模块功能。

## 3.1 系统设置模块

本模块是 OLAP 设计器的一部分,主要包括:用户管理,数据库连接信息设置和元数据管理,完成系统的初始设置和初始准备。

## 3.2 模型管理模块

这里的模型是指多维分析模型。用户可以根据分

<sup>\*</sup> 本文得到八六三“基于曙光系列计算机的大型信息系统”(863-306-ZD-07-4)和国家自然科学基金“保险信息处理与精算数学理论和方法”(19831020)重点项目的资助。刘瑶 硕士生,主要研究方向为大型数据库和网络工程。孙玉芳 研究员,博士生导师,主要研究方向为中文信息处理、系统软件、大型数据库与网络工程。

析需要创建各种分析模型,例如:帐务分析模型和分析树分析模型。由于在系统设置中对元数据进行了设置,用户在创建分析模型时完全可以使用习惯的术语,由系统完成术语与元数据的对应。该模块还负责将用户定义的分析模型转换成文件的形式保存起来,必要时可以修改/删除已建立的模型。

### 3.3 分析树管理模块

分析树管理模块负责创建和构造分析树。每创建一棵分析树,系统赋予其一个唯一的 id 号。系统采用 TreeView 的形式显示分析树,用户可以用拖拽的方式方便、快速地构造一棵分析树。每棵分析树可以作为分析模型中的一维,但一个分析模型中只能有一个分析树维。在构造分析树的同时对树结点要描述其属性,如:变量属性、计算属性等。树结点可以有零到多个属性。必要时可以修改/删除已有的分析树。

### 3.4 OLAP 分析模块

OLAP 分析模块是整个系统中最重要模块,该模块负责执行各种分析操作。在进入 OLAP 分析模块之前,用户必须首先定义分析模型。在进入 OLAP 分析模块之后,用户从已定义的分析模型中选择一个分析模型。设计器与分析器之间用文件作为桥梁,传递分析模型的结构。

OLAP 分析模块除要完成各种基本操作外,还可以实现灵活的粒度维和分析树维新增的操作功能。对于灵活粒度维需要根据设计器中的定义随时更新模型结构并对影响模型结构的漂移操作进行特殊处理,显示正确的分析结果。对于分析树维则需要根据分析树的结构、树结点的属性显示相应的分析结果。在分析树维上的操作仅限于钻取操作,但是对非分析树维上的操作没有限制。

我们将 OLAP 的相关操作,诸如钻取、旋转、切片、漂移等组合在同一个操作界面下,在每个操作时刻以菜单是否可操作的形式显式地提醒用户下一步可以做哪些操作,不可以做哪些操作。也就是说由系统自动判断可执行操作,这样使得分析动作更直观而且简单。

在分析过程中,用户可以通过导航图随时定位当前所处的位置,决定下一步操作。如果分析模型包含了灵活粒度维,导航图将随着粒度结构的变化随时更新。如果分析模型包含了分析树维,则用户可以直接在导航图上操作,同时在分析结果显示区显示结果。这时导航图和分析结果显示区在系统内部存在一个对映关系,使得无论用户在导航图上操作还是在分析结果显示区内操作都可以达到同样的目的。

### 3.5 图表分析模块和函数功能模块

图表分析和函数都是 OLAP 的辅助分析功能。用图表的方式表达分析结果比直接用数字表达分析结果

要生动、直观。在用户分析的过程中,随时可以将当前分析数据转换成图表表示方式,系统提供了丰富的图表形式供用户选择。

用户有时需要对当前数据进行简单的加工处理,例如:显示最大/最小值,求占用百分比、求均值等等。系统内部提供了一些预定义的函数,用户可以直接使用。当预定义的函数中没有满足需求的函数时,系统还允许用户适当地定义一些自定义函数。

## 4. 设计器实现

设计器实现的主要功能是管理用户、管理分析元数据、构造分析逻辑模型,为分析做准备。

### 4.1 设置数据库连接信息

由于本系统采用的是客户机/服务器结构,因此在分析之前,前端分析程序需要与后端数据库建立连接。系统允许设置多个连接信息,连接到不同的数据库,由用户在分析之前确定连接哪个数据库。

数据库连接信息保存在文件 profile.ini 中,文件格式详见文[3]。

### 4.2 管理元数据

这里的元数据包括:数据列、维、分析树结点。

1) 数据列管理是将数据列的英文名与中文名建立对应关系,使得用户在定义多维分析模型时可以直接使用中文名,使定义更加直观。数据列一般都作为多维分析模型中的粒度。

2) 维管理就是定义所有可能的维。

3) 由于分析树的大部分非计算结点(非计算结点是指所有不是通过子结点计算而来的结点)都来自于数据库,而且分析树之间的非计算结点也存在交集,因此将这些非计算结点统一管理起来不仅能够使得分析树的定义变得简单,而且也易于维护这些结点。

非计算结点有两种属性:select 属性和 label 属性。具有 select 属性的结点必须确定相应的取数函数<sup>[2]</sup>。具有 label 属性的结点只起到一个标识作用,这种结点没有相关的数据,在分析树上起到分类的作用。系统为所有这些结点分配一个唯一的 id 号。

### 4.3 构造分析树

按分析需求组织构造分析树,分析树结点可以是前面已经定义的结点,也可以是新定义的结点,系统为每棵分析树分配一个唯一的 id 号,为每个新定义的结点分配一个唯一的 id 号。对于新定义的树结点需要定义结点属性,可能的结点属性包括:select 属性、label 属性和 compute 属性。select 属性和 label 属性与上面所讲的非计算结点的属性相同,compute 属性是指该结点的值是由于结点的值通过计算得到的,该属性必须定义计算表达式,子结点在表达式中用其 id 号表

示。例如:  $1-1=4-2+5-2$ , 表示结点 1 的第一个数据项的值等于结点 4 第二个数据项的值乘以结点 5 第二个数据项的值。

#### 4.4 建立多维分析逻辑模型

确定多维分析模型的结构, 包括: 模型中有多少个维, 每一维的粒度构成, 是否有特殊维(灵活粒度维或分析树维), 如果有灵活粒度维则必须确定粒度变化参数表, 如果有分析树维则必须给出分析树 id 号。每一个分析模型中最多只能有一个特殊维。

由于系统对维、粒度、分析树都是统一管理, 因此用户只须通过选择和拖拽操作即可按分析需要建立合适的多维分析模型。

#### 4.5 定义初始显示

初始显示是分析的起点, 用户从初始显示开始进行各种分析操作: 钻取、切片、旋转等。在初始显示中需要: (1) 确定显示的格式是交叉表还是简单列表; (2) 确定显示的数据列; (3) 确定显示数据的前提条件(如: 时间、单位)。对每个多维分析逻辑模型都必须定义初始显示。

设计器和分析器之间是通过报告定义文件 report.ini 相互联系的。设计器将用户定义的分析模型保存在 report.ini 中, 分析器则通过读取该文件, 用多维数组重新构造分析模型。report.ini 的格式详见文[3]。

### 5. 分析器实现

分析器主要是在原来 OLAP 工具的基础上作改进, 包括: 初始显示、灵活粒度操作和分析树操作。

#### 5.1 初始显示

初始显示需要改进的是允许初始显示带有多个前提条件。实现时, 在文件 report.ini 中增加了初始显示条件和初始显示条件取值两项。用户在分析器中打开一个报告时, 系统读取这两项的值, 加入到 SELECT 语句的 WHERE 字句中, 然后再执行 SELECT 语句。同时, 每个初始显示条件都以下拉列表框的形式显示给用户。

需要注意的是灵活粒度维中的灵活粒度不允许作为初始显示中的前提条件。

#### 5.2 灵活粒度操作

灵活粒度操作实现的关键是运行中变更粒度结构和强制漂移。

1) 运行中变更粒度结构 在 report.ini 中, 对包含灵活粒度维的分析模型有两个重要的参数:

决定粒度变化的数据列=数据列个数 q, 表名-列名 1, 表名列名 2, …… 表名列名 q  
灵活粒度维粒度变化参数表=参数表名称  
(其中维 1 是灵活粒度维)

· 决定粒度变化的数据列。灵活粒度维的粒度变化取决于一个或多个字段的取值, 例如: 一级科目以后的科目划分取决于核算单位和一级科目。该参数就是列举所有影响灵活粒度变化的字段, 以便在分析操作中知道何时改变粒度结构。

· 灵活粒度维粒度变化参数表。该参数表决定粒度如何变化。在参数表中必须包括所有决定粒度变化的数据列, 例如: 某维结构中, 维 2 的粒度 1 和维 3 的粒度 2 决定灵活粒度维 4 的粒度变化, 则参数表的结构必须是(粒度 1, 粒度 2, 粒度结构)。

该参数表可以是应用系统中已经存在的, 也可以是为分析而建立的。在报告定义中必须给出的是参数表的名称。

系统在运行中通过查询灵活粒度维粒度变化参数表可以知道粒度变化情况, 及时更新灵活粒度维的粒度结构。变更粒度结构的同时更新导航图。

2) 强制漂移 由于粒度的灵活变化, 必然给漂移操作带来一定的限制。在限制有效性漂移操作中, 仍然是决定粒度变化的数据列和灵活粒度维粒度变化参数表这两个参数起重要作用。

漂移操作实际上是改变某一列的取值。如果该列不是决定粒度变化的数据列, 则是正常漂移。但如果该列是决定粒度变化的数据列, 则灵活粒度维的粒度结构有可能发生变化, 是否发生了变化可以通过查询灵活粒度维粒度变化参数表得知。如果该漂移操作确实使得粒度结构发生了变化, 就需要进行强制漂移, 改变粒度结构(即改变多维数组的某些取值), 重新取数, 显示正确结果。

#### 5.3 分析树操作

实现分析树操作的关键在于显示变量随结点变化而变化、中间结点的计算和导航图可操作。

1) 显示变量随结点变化而变化 变量是数据的实际意义, 即描述数据“是什么”。在原 OLAP 工具中, 维和粒度的取值都是对变量的描述。显示的变量是库表中的一个数据列或一个计算列, 粒度取值无论如何变化, 都不影响变量。但是在分析树维中, 变量是与树结点相关的, 不同的结点可以对应不同的变量。结点的变量可以是子结点的计算结果, 也可以是数据库中某个数据。因此变量成为树结点的一个属性。

在分析树定义中, 每个结点都有属性定义, 在属性定义中决定结点显示的变量。在对分析树操作时, 需要获取当前结点的属性描述, 然后决定如何显示变量及变量的值。

由于有的结点的变量是其子结点的计算结果, 只有在知道子结点的变量取值后才能计算父结点的值, 这就引出了中间结点计算的问题。

2)中间结点计算 中间结点的计算是一个实现的难点,实现中间结点的计算可以有两种方法:一是在分析到某个结点时临时计算,二是在分析之前计算所有结点。

如果采用临时计算,对于那些计算结点每次都要从叶结点开始,从下到上依次计算。这样使得一是效率不高,二是容易造成大量的重复计算。

如果采用在分析之前计算所有结点,并保存计算结果,则在分析时效率很高,因为结点的变量值是现成的,用户等待的时间很短,但是这种方法的缺陷在于如果改变分析树的任意一个前提条件,就必须重新计算所有结点的值,这时用户需要等待很长一段时间。

我们在实际实现时综合了以上两种方法。在选定一个报告并给出初始条件(时间,单位等)后,系统将所有标志为'select'的结点都从库表中获取其值,并保

存。在分析操作时,对当前的结点临时计算,同时保存该结点所有下层结点的计算中间结果。这样,当分析下层结点时就不必重新计算。如果初始显示是从根结点开始,就必须计算所有结点。采用这种方法,需要记录当前计算值的前提条件,当计算前提条件不变时:①访问已计算的结点时,无须重新计算,直接显示变量值;②访问未计算的结点时,如果相邻下层结点已经计算过则直接利用这层变量值计算,否则须从下一层尚未计算的结点开始计算并保存计算路径上的每个结点的计算结果,直到当前访问的结点。以下具体说明该算法。

例如:对杜邦分析法<sup>[4]</sup>可以构造如下:(时间维,单位维,分析树维) 时间维(年度,季度,月份) 单位维(核算单位,基层单位) 分析树维的结构如图2所示。假设初始显示结点是销售净利润率。

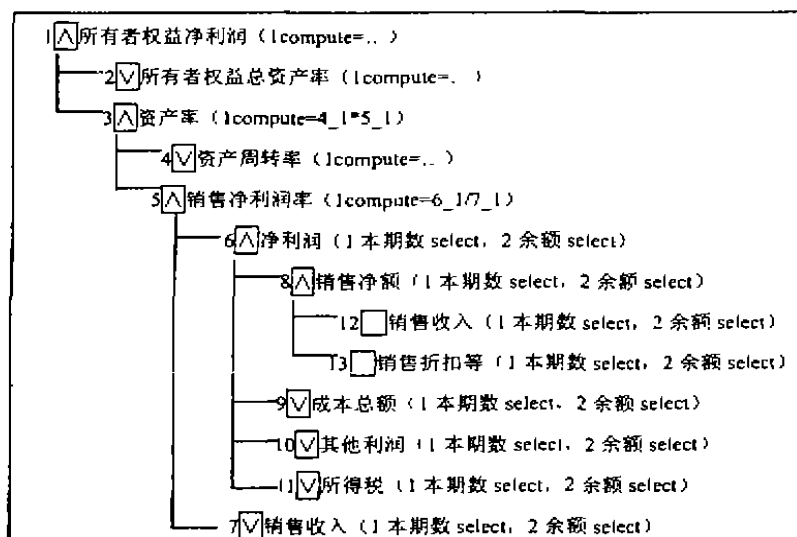


图2 分析树维的结构

①用户选定报告后须首先确定时间和单位,然后系统根据时间和单位这些前提条件计算每个标志为'select'的结点数据项,即构造并执行select语句,保存结果,同时记录当前分析的前提条件,即其他维(除分析树维以外的所有维)粒度的取值。

②由于销售净利润率结点数据项标志为'lcompute',说明该结点的第一个显示数据项的值是由下一层结点通过计算得到的,系统找到其下一层结点然后计算。计算公式 $lcompute=4\_l * 5\_l$ 表示该结点的第一个数据项的值等于结点4第一个数据项的值乘以结点5的第一个数据项的值。结果被保存。

③当分析销售净利润率的所有子孙结点时:a)首先比较当前其他维粒度取值与结点中已保存的其他维粒度取值,如果相同,说明其他维粒度的取值未发生变

化。由于这些结点都已经计算过,并且保存了结果,所以直接返回各结点数据项的值即可;b)如果不同,说明其他维粒度的取值发生了变化(进行过漂移操作),则需要重新计算该结点。如果结点的数据项标志为select,则根据当前其他维粒度取值重新构造并执行select语句;c)如果结点的数据项标志为compute,则检查下一层结点的condition是否与当前一致,如果一致则直接利用下一层结点数据项的结果计算该结点数据项的值;d)如果不一致,则需要重新计算所有下一层结点,这时令下一层结点为当前结点,回到a。

#### 5.4 导航图可操作

由于分析树上的结点实际上是枚举结点,对于分析树的分析操作,为了使用户使用方便,应允许用户在

(下转第40页)

口。接收网络管理应用的数据访问请求,通过数据管理控制选择适当的数据采集方案。

总体说来,rNetDB的功能可以分为采集和管理两个部分。由应用接口封装为面向管理应用的虚拟网络管理数据库。数据采集负责和底层管理协议之间的通信和采集到的MIB数据与关系数据库之间的格式转换;进行数据管理的主要目的是提高数据库的使用性能。数据采集和管理之间以事件方式交互,实时性能良好。

**结束语** rNetDB是一个基于关系数据库的网络管理数据采集和管理模型。它的主要特点是:

①采用关系数据库管理网络数据,提高了管理水平并使应用接口标准化;

②区分MIB数据的特点,采用不同的采集和存储方案,提高数据管理的性能;

③在MIB对象注册的基础上实现MIB数据的转换和定位,解决了MIB-关系数据库表的映射问题;

④对海量历史数据采用缓冲和压缩,改善系统性能;

⑤数据采集和管理应用之间以事件作为同步机制,保证数据具有良好的实时性。

在网络管理系统的开发过程中,我们深深体会到实现合理高效的数据采集和管理对于网络管理系统的

重要性。目前,SuperDomain98系统已成功在CERNET主干网上试运行近一年,表现出良好的性能。实践证明,rNetDB是行之有效的网络管理系统数据采集和管理的解决方案。

## 参考文献

- 1 Yemini Y, Yemini S. Modeling Management Semantics. Second IEEE Network Management and Control Workshop, New York, September 1993
- 2 Wolfson O, Sengupta S, Yemini Y. Managing Communication Networks by Monitoring Databases. IEEE Transactions on Software Engineering, 1991, 17(9)
- 3 Ilsoo A. Database Issues in Telecommunications Network Management. SIGMOD Conference 1994
- 4 杨家海,张俊波. 网络管理系统 NetView/6000 结构及应用分析. 计算机工程与应用, 1999(1)
- 5 Martin Lee S, Mark F, James D, Jeffrey C. A Simple Network Management Protocol, RFC1157, 1990
- 6 Yang Jiahai, Wang Peiyu, Wu Jianping. A Scalable, Web-based Architecture for Hierarchical Network Management. IEEE GLOBECOM'99, Brazil, Dec. 1999
- 7 杨家海,李星,吴建平. 基于Web/Java的网络管理系统 SuperDomain98的设计与实现. 清华大学学报(已录用,待发表)

(上接第65页)

导航图上操作,直接跳转到目的结点。用户操作界面见图[3]图4-7。

用户在该界面下,在导航图上可以选择任意感兴趣的结点,结点的数据项显示在右边的分析操作区中。同时用户在分析操作区中可以对结点进行上钻,下钻及漂移等操作。

• 在导航图上操作 由于在存储分析树时,保存了所有结点各自结点id。结点id = treeview的itemhandle值,结点id在一棵树中是唯一的。用户选择任意结点时,系统可得知该结点的id,根据该id,在可找到对应结点的所有信息,然后将该结点显示在分析操作区中。

• 在分析操作区中操作 向下钻取实际上是显示子结点,向上钻取实际上是显示父结点,而漂移操作是重新计算结点,与上一小节中间结点计算相同。存储分析树时,对每个结点都保存了其父结点和子结点的id,因此对上钻、下钻操作很容易定位父结点、子结点。

**小结** 近几年来OLAP以其符合人们思维的多维结构越来越成为强有力的分析工具,而且我们已经有了一个比较完善的OLAP工具。

由于财务分析有其自身的特殊性,经过分析和探讨,我们以原OLAP工具为原型,进一步设计并实现了通用财务分析器。目前该系统已在中国人寿保险公司几个分公司使用,用户反映良好。当然,系统还有待进一步提高稳定性,查询优化和运行效率。

## 参考文献

- 1 Codd E F, et al. Providing OLAP(On-Line Analytical Processing) to User Analysts. PC WORLD, 1993. 9
- 2 刘瑶,孙玉芳. 通用报表生成器的设计与实现. 计算机科学, 1999. 5 待发表
- 3 刘瑶. 通用财务分析器的设计与实现. [硕士论文]. 中国科学院软件研究所, 1999. 5
- 4 谢忠阳,刘瑶,左春. OLAP在财务分析中的应用. 计算机工程与设计, 1999. 5 待发表