文章编号:1001 - 2486(2002)06 - 0082 - 05

基于 GMM 的数据仓库管理与维护*

戴超凡,邓 苏,杨 强,陈文伟,刘青宝
(国防科技大学人文与管理学院,湖南 长沙 410073)

摘 要:元数据管理系统是构建、管理、维护和使用数据仓库系统的核心部件。元数据管理关键在于构建一个全面的、可扩展的元数据模型,表示各种类型的元数据。本文提出了一个通用的元数据模型 GMM,该模型可以有效支持数据仓库的管理和维护,如用户视图管理、个性化服务、增量刷新、数据志跟踪等。

关键词:数据仓库;元数据;元数据模型;元数据库
中图分类号:TP311.132 文献标识码:A

Management and Maintenance of Data Warehouse Based on GMM

DAI Chao-fan, DENG Su, YANG Qiang, CHEN Wen-wei, LIU Qing-bao
(College of Humanism and Management, National Univ. of Defense Technology, Changsha 410073, China)

Abstract: Metadata management system is the core of construction and management and maintenance and use data warehouse system. The keys of metadata management lie in the construction a comprehensive and extended metadata model which describes all kinds of metadata. This paper presents a General Metadata Model(GMM) which can support management and maintenance of data warehouse effectively, such as management of the user view, personalized service, incremental refreshment, data lineage tracing, etc.

Key words: data warehouse; metadata; metadata model; metadata repository

元数据管理系统是构建、管理、维护和使用数据仓库系统的核心部件,是技术人员开发与维护数据仓库的蓝图,是终端用户导航数据仓库和定位有用信息的路标[1~3]。数据仓库系统中元数据管理的关键在于两个方面[2]:(1)构建一个全面的、可扩展的元数据模型,表示各种类型的元数据。元数据模型又称为信息模型[4]、元数据方案[5,6]或元模型[2,7],它表示和存储所有元数据的物理数据库方案;(2)建立柔性的管理机制,实现有效的元数据管理。

Staudt 详细介绍了元数据在数据仓库系统中的作用、标准化工作和研究项目[1]。比较有影响的数据仓库元数据标准是 MDC 的 OIM[4]和 OMG 的 CWM[7]。Vetterli 对 OIM 和 CWM 进行了详细的比较[8,9]。OIM 开始是软件工程领域的元数据表示标准,扩展后的 OIM 包含信息系统开发的所有阶段,支持多种计算技术,如 CASE、组件、应用程序、Intranet、数据库、数据仓库[3,4,8~10],而 CWM 的主要目标是便于分布、异构环境中在数据仓库工具和元数据库之间交换元数据[7~9]。由于两个标准都基于相同的底层标准,即 UML 和 XML,这为两个标准的合并提供了可能[1,11]。

数据仓库研究项目也提出了很多元数据模型。Stöhr 基于 UML 对技术元数据和语义元数据进行一体化建模和管理[5]。Macro 给出了一个基本的元数据模型,可以用来检验和评估元数据模型和元数据库工具[2]。Sachdeva 设计的元数据模型增加了查询统计、转换历史纪录、用户管理等[6]。Vavouras 提出了一个处理增量刷新的元数据模型,可以对结构和操作进行映射[12,13]。罗昌隆提出了一个较易实现、结构清晰的元数据参考模型[14]。但是,目前还没有一个通用的元数据模型能够被大多数软件商、用户和研究人员所认可和采纳,而且缺乏一种系统的元数据管理方法和结构。上述元数据模型都不能完备、有效地描述和控制数据仓库:

* 收稿日期:2002 - 05 - 06
基金项目:国家部委基金资助
作者简介:戴超凡(1973—),男,博士生。

(1) 除文献[14]之外,都没有直观体现数据仓库的结构特性;
(2) 没有对管理功能进行建模,尤其是没有对用户视图管理、个性化服务等建模;
(3) 除文献[12,13]之外,没有考虑增量维护问题;
(4) 除了 OIM 和 CWM 支持粗粒度的数据志之外,其它都不支持数据志;
(5) 除 OIM 和 SMART 之外,其它元数据模型都不具备自描述能力。

1 通用的元数据模型 GMM

元数据模型用于存储数据仓库系统中基本的元数据,包括业务主题、仓库数据模型、源到目标的数据映射和语义转换、源方案、ETL 统计、查询统计、用户管理、归档和清理规则、数据志跟踪方案等[1~3,14]。本文使用 UML 建立了一个通用的元数据模型 GMM (General Metadata Model),主要用于支持数据仓库管理与维护和数据志跟踪,对比较成熟的关系数据库方案的建模进行了简化。GMM 由自描述模型和描述模型两部分组成,相应的模型信息存储在基于关系存储模式的元数据库中,是元数据库中最核心的元数据,是 GMM 自维护、自描述和元数据管理的基础,也是 GMM 可扩展性的保证。

1.1 自描述模型

自描述能力是衡量元数据模型的一个重要指标[4,15],可以实现元数据模型及其实例(元数据)的一致管理,便于元数据模型→元数据→数据资源的三级导航和资源定位[15]。GMM 是一个可自描述的模型,它涉及到的建模元素包括:类、关联、关联类、继承类、属性、主属性、关键主属性、外主属性、关联度等,如图 1 所示。

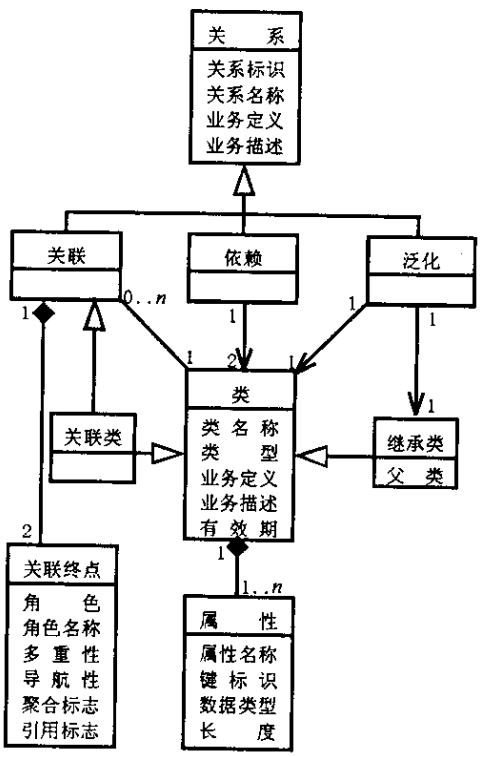

图 1 GMM 的自描述模型

Fig.1 Self-descriptive model of GMM

GMM 中的建模元素与 UML 中的相应概念在语义上是一致的,其中:

(1) 类是最底层、最基本的元素,类可以表示一个关系表、一个记录类型或者一个商业概念。类的结构通过类的属性来描述,类可以与其自身或其它类关联;

(2) 类的主属性可以惟一标识一个类的属性组,其中类的固有属性称为关键主属性,来自于与其关联的类的主属性称为外关键属性;

(3) 类与类之间可以具有关联、组合、依赖和继承这几种关系之一;

(4) 关联的主属性由参加关联的源类和目的类的主属性(需要消除重复)组成,如果关联还具有私有属性,则必须建立关联类;

(5) 一个类是另一个类的‘部分’,并且它们一起消亡,那么它们构成组合关联;

(6) 一个类可以依赖于 0 到多个类;

(7) 继承类从一个或多个父类继承而来,继承了所有父类的所有属性。而关联类一般只与两个类具有关系,并且关联类只‘继承’源类和目标类的主属性。

1.2 描述模型

GMM 的描述模型简化了对关系数据库方案的描述,侧重于对数据仓库的管理和维护进行建模。GMM 的描述模型又包括三个子模型:

(1) 结构子模型:层次地刻画了数据仓库系统的结构特性:用户→主题→目标表→转换→源数据,如图 2 所示;

(2) 转换子模型:层次地刻画了数据仓库系统中数据转换处理的结构特性:转换图→转换→属性映射,还描述了转换(任务)之间的逻辑顺序,并且对批量加载和增量刷新过程进行一致建模,如图3所示;

(3) 数据志方案子模型:层次地刻画了数据仓库系统中数据志跟踪方案的结构特性:转换图→转换树→转换序列→转换↔(弱)逆转换→属性映射↔(弱)逆映射,记录转换和属性映射的(弱)逆信息,包括转换和属性映射的可逆程度、结果精度、(弱)逆规则、延续跟踪性标志等信息,支持细粒度的、实例属性级的数据志跟踪,如图4所示。

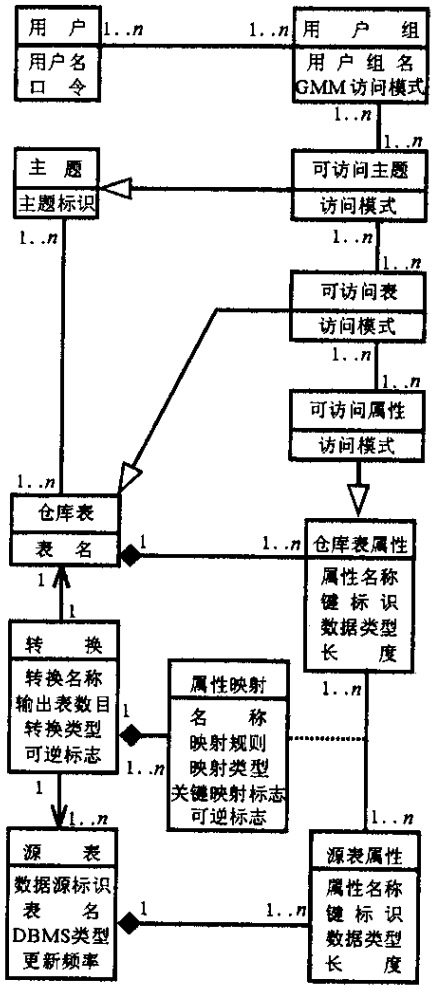

图2 GMM描述模型中的结构子模型

Fig.2 Structure sub-model of descriptive mode of GMM

2 基于GMM的数据仓库管理与维护

数据仓库系统获取、共享和管理的元数据主要有两个目的:

(1) 作为一致的描述性信息:描述系统的结构特征和静态特征;

(2) 作为特定工具的控制性信息。工具将其静态信息或程序的部分运行逻辑作为控制性元数据存储在程序之外的元数据库中。运行时,读取这些元数据并动态绑定。程序配置和需求变化时,可以很方便地改变元数据而无需修改程序,更不需要重新编译程序,这样极大地增强了系统的柔性,可以实现数据仓库管理和维护的(半)自动化处理。

GMM所包含的元数据类型不仅仅提供了基本的结构描述能力,而且提供了很强的控制能力,在此基础上可以灵活地实现用户视图管理、个性化服务、增量维护、数据的归档和清理等数据仓库管理与维护功能,还可以实现数据志的跟踪。同时,由于对技术元数据和语义元数据进行一体化建模,因此,可以为用户提供相应的技术视图或业务视图,还可以方便地实现两种视图之间的过渡和导航。图5给出了基于GMM的数据仓库管理与维护结构。

2.1 用户视图管理

在数据仓库系统中,用户视图管理的主要目标是将用户查询简化到该用户可以访问的数据集上,主要包含两部分:

(1) 用户视图管理器:为每个用户(组)定义单独的、可访问的用户视图框架(包括访问对象、访问规则和访问模式等),即简化的、定制的数据仓库。

(2) 查询管理层:活动范围从元数据层延伸到数据服务器,其主要任务就是接受用户查询并检查是否允许执行。查询管理层将用户查询透明地简化到其授权的'定制的'数据仓库中去,同时还要给用户一个'整体'印象——他看到的内容就是整个数据仓库的内容。这样用户就不会感觉到约束和限制,也不会试图去访问没有授权的数据。

2.2 个性化服务

现有的数据仓库系统在服务模式方面存在着明显的缺陷,服务质量比较低。提高服务质量的关键在于从被动接受用户请求转化为主动感知用户的信息需求,实现数据仓库对用户的主动信息服务,实现"信息找人,按需服务"[16]。

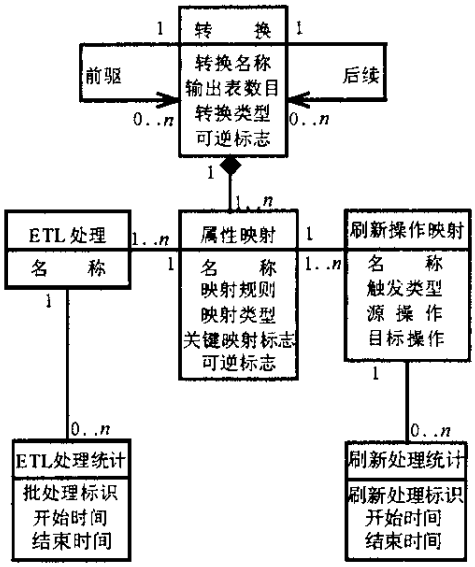

图 3 GMM 描述模型中的转换子模型

Fig.3 Transformation sub-model of GMM

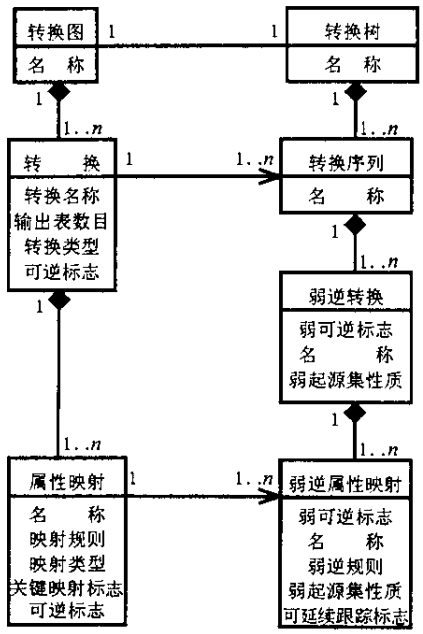

图 4 GMM 描述模型中的数据志子模型

Fig.4 Data lineage sub-model of GMM

GMM 描述了用户的信息需要、用户的兴趣领域或主题，用户的访问方式等用户概貌信息，个性化服务的内容是安全管理器定义和描述的‘简化’的数据仓库，用户组和用户是用户视图管理器指定的用户组和用户。因此，可以将个性化服务与用户视图管理结合起来，为用户提供一体化的、个性化的访问接口。

2.3 联机增量维护

当数据源中的数据改变时，数据仓库中的数据也要及时更新，使数据仓库随时处于一致的可用状态。联机增量维护是解决这些瓶颈的关键，其核心是增量刷新，这是一个非常复杂的过程，包括对数据源进行监视、抽取、转换、集成、清洗、导出新数据、记录数据志、更新到数据仓库。

GMM 对源到目标的映射以及初始批量加载和增量刷新过程进行了一致建模，为增量刷新提供全面的支持，支持各种监视技术，柔性地描述和驱动刷新过程。对刷新过程建模主要是根据数据源方案信息以及源到目标的转换和映射等结构信息，定义增量刷新的操作映射、触发类型、预处理方法、刷新规则等。与刷新过程相关的工具主要包括：

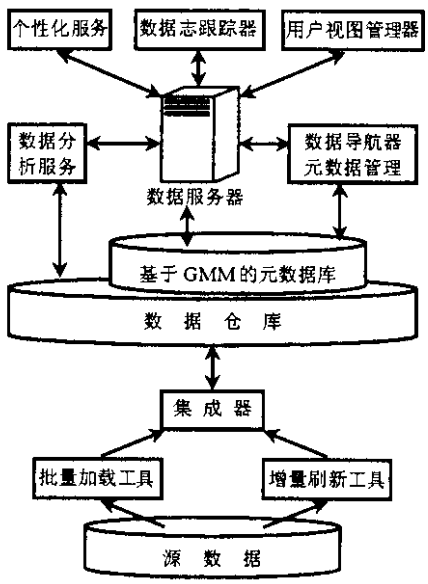

图 5 基于 GMM 的数据仓库管理与维护结构

Fig.5 Structure of DWS based on GMM

(1) 监视器：监视数据的变化及其操作类型，将变化的数据从数据源抽取出来；

(2) 封装器：经预处理后，将抽取出来的数据转换成一种‘统一’的数据仓库数据表示形式。封装器对基于事件和基于状态的数据需要分别处理；

(3) 集成器：对封装后的数据进行综合集成，然后协调地增加到数据仓库中去。

2.4 数据的归档和清理

数据仓库是动态的系统，不仅仅反映在监视与刷新上，同时还反映在数据资源的归档与清理上。数

据资源都有其有效的生命周期,一旦超出了这个生命周期就失去意义。

根据 GMM 中定义的有效的生命周期和访问频率,清理和归档工具可以评估或直接指定数据资源的有效期,对很少使用和过期的数据资源或直接删除,或迁移到二级存储器,并保持元数据的同步。这样可以减少数据的存储量和网络通信量,节省大量存储资源和网络资源,同时能为这些数据资源提供自动、透明的访问。

2.5 数据志跟踪

在数据仓库系统中,给定一个数据仓库中的数据项/集,获取该数据项/集精确的历史沿革,即数据项/集从获取、转换、集成到现状这一完整生命周期的有关描述,称为数据志跟踪。数据志跟踪技术逐渐成为数据仓库环境中的热点问题,不仅可以支持更全面、更深入的数据分析,还可以帮助技术人员验证源、清洗和转换过程的正确性,提高数据仓库质量。数据志跟踪器主要包含两部分:

(1) 数据志方案注册器:注册 GMM 中数据志方案子模型中的有关信息;

(2) 数据志跟踪引擎:基于属性映射的弱逆和验证的方法,逐步求解精准的或单纯的起源集。

3 结论

在分析现有元数据标准和元数据模型的现状的基础上,本文提出了一个通用的元数据模型 GMM,GMM 简化了关系数据库方案的建模,侧重于对数据仓库管理与维护以及数据志跟踪进行建模,其主要优点是:(1) 层次地刻画了数据仓库系统的结构特性;(2) 一致描述了初始批量加载和增量维护过程;(3) 支持数据志跟踪;(4) 支持用户视图管理和个性化服务;(5) 具有自描述能力。

但是,GMM 也存在很多不足之处,如对多维数据模型的描述能力很弱;语义元数据只具备最基本的描述能力,不具备语义推理能力;面向关系数据库,目前不能描述层次、网络、对象和多维数据库。

基于 GMM,本文给出了数据仓库管理与维护结构,阐述了管理与维护工具应用 GMM 中的描述性元数据来(半)自动完成任务和柔性配置的基本情况。

参考文献:

[1] Staudt Martin, Vaduva Anca, Vetterli Thomas. The Role of Metadata for Data Warehouse [R]. http://www.ifi.unizh.ch/techreports/TR-1999.html, 1999.

[2] Macro David. Building and Managing the Metadata Repository—A Full Lifecycle Guide [M]. John Wiley & Sons Inc., 2000.

[3] 戴超凡,陈文伟,邓苏等.数据仓库中的元数据技术研究 [J].计算机工程与应用,2001,37(14):85-87.

[4] Metadata Coalition. Open Information Model Version 1.0 [S]. http://mdcinfo.com.

[5] Stöhr T, Müller R, Rahm E. An Integrative and Uniform Model for Metadata Management in Data Warehousing Environments [C]. DMDW'99, 1999:1-16.

[6] Sachdeva Satya P. Metadata Architecture for Data Warehousing [EB]. http://www.dmreview.com/master-sponsor.cfm?NavID=55&EdID=664, 1998.

[7] OMG. Common Warehouse Metamodel Specification [S]. http://www.omg.org, 1999.

[8] Vetterliy Thomas, Vaduvaz Anca, Staudty Martin. Metadata Standards for Data Warehousing: Open Information Model vs. Common Warehouse Metamodel [C]. SIGMOD Record, 2000, 29(3): 68-75.

[9] Vetterli T. A Comparison of OIM with CWM [EB]. http://www-ai.cs.uni-dortmund.de.

[10] 戴超凡,邓苏,陈文伟等.开放信息模型研究[J].计算机工程与应用,2001,37(1):14-16.

[11] Group Gartner, Blechar M. OMG's Common Warehouse Metamodel Specification [EB]. Research Note 2000, 7, 28.

[12] Vavouras Athanasios, Gatziu Stella, Dittrich Klaus R. SIRIUS: An Approach for Data Warehouse Refreshment [R]. http://www.ifi.unizh.ch, 1998.

[13] Vavouras Athanasios, Gatziu Stella, Dittrich Klaus R. Modeling and Executing the Data Warehouse Refreshment Process [R]. http://www.ifi.unizh.ch, 2000.

[14] 罗昌隆,黄梓龙.数据仓库的元数据模型的探讨[J].南京邮电学院学报(自然科学版),2000,20(2):80-82.

[15] Vaduva A, Dittrich K R. Metadata Management for Data Warehousing: Between Vision and Reality [C]. In Proc. of IDEAS'01, France July 2001.

[16] 李勇.智能检索中基于本体的个性化用户建模技术及应用[D].国防科技大学学位论文,2002.