文章编号: 1007- 144X(2005)01- 0026- 05

数据仓库中元数据管理技术

王东龙, 陈 建, 李茂青

(厦门大学 计算机与信息工程学院, 福建 厦门 361005)

摘 要: 元数据是数据仓库实现和管理的灵魂, 也是实现数据仓库系统的一个难点。首先介绍了元数据的定义、作用和分类; 然后讨论了数据仓库系统中元数据的标准化情况, 着重探讨了元数据的管理策略以及元数据体系结构。

关键词: 数据仓库; 元数据; OIM; CWM

中图法分类号: TP31 文献标识码: A

1 引 言

元数据的概念并不新鲜, 但是元数据在数据仓库环境中的作用和重要性是不言而喻的。元数据是数据仓库中的一个重要组成部分, 元数据管理系统则是构建、管理、维护和使用数据仓库系统的核心部件。如果一个数据仓库中没有元数据, 那么用户就不知道如何进行分析。用户必须首先对数据仓库进行各种试探才能确认其中有哪些数据和没有哪些数据, 这样就浪费了大量的时间, 而且不能保证一定能找到正确的数据。因此元数据及其管理对于数据仓库的成功实施是至关重要的。

目前国内关于数据仓库元数据的研究主要在元数据定义及分类、元数据标准、基于 XML (XMI) 的元数据交换、元数据管理结构、元数据模型等领域。关于元数据的定义众多的文章均有介绍[1~13], Inmon W H 关于元数据的定义[10] 是目前被引用最多的, 吕波等提出了广义元数据和狭义元数据[6]。关于元数据的分类有很多, 胡颖峰等的研究从多个视角展示了元数据[1, 3, 4, 6, 9], 对元数据的分类研究有助于加深对元数据的认识, 更好地集成元数据。到目前为止元数据业界标准还没有最终定型, 在 2000 年 9 月 MDC 并入 OMG 后, 最有可能形成统一标准的是 OMG 的 CWM。戴超凡等详细介绍了 OIM 和 CWM[1, 8, 13]。而关于元数据的交换的研究大部分体现在对关系数据库中, 这主要是因为构建数据仓库所需面对的遗留系统大部分仍是基于关系型数据库的, 关于其他数据源中的元数据的集成和交换的研究比较少。在元数据管理方面, 曹蓟光等详细介绍了几种元数据管理策略并进行了比较[2, 4, 7]。笔者介绍了元数据的一些基本知识以及前人的结果, 分析了一些高级元数据策略以及其所对应的元数据体系结构, 并展望了元数据未来的发展趋势。

2 基本知识

2.1 元数据的定义

人们通常将元数据定义为 "关于数据的数据[10] "或者 "描述数据的数据 "。元数据是指来自企业内外的所有(包括软件和其他介质中含有的) 物理数据和(员工和各种媒介中含有的) 知识, 包括物理数据的格式、技术和业务过程、数据的规则和约束以及企业所使用数据的结构。元数据其实就是知识, 包括系统、业务和市场的知识。

2.2 元数据的作用

元数据类似于指向数据仓库内容的索引, 处于数据仓库的上层, 并且记录数据仓库中对象的位置, 是内部技术人员开发与维护数据仓库的蓝图, 是业务终端用户导航数据仓库以及定位有用信息的路标[2]。数据仓库系统获取、共享和管理元数据主要有 2 个目的。

收稿日期: 2004- 09- 08.
作者简介: 王东龙(1980-), 男, 福建泉州人, 厦门大学计算机与信息工程学院硕士研究生.

(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) 作为控制性信息, 控制并配置特定工具和进程运行, 实现数据仓库管理和维护的(半) 自动化处理。

在数据仓库系统中, 元数据机制主要支持以下 5 类系统管理功能: 描述哪些数据在数据仓库中; 定义要进入数据仓库中的数据和从数据仓库中产生的数据; 记录根据业务事件发生而随之进行的数据抽取工作时间安排; 记录并检测系统数据一致性的要求和执行情况; 衡量数据质量。

元数据在数据仓库系统中发挥着至关重要的作用, 如元数据可以用于集成各类复杂繁多的信息; 元数据定义的语义层可以帮助最终用户理解系统中存储的数据; 元数据可以支持需求动态变化, 系统各项表现(界面等) 的灵活性; 元数据可以提高和保证数据的质量; 元数据可以支持多种工具的开发应用; 元数据可以提高系统的安全性; 元数据可以提高系统的智能性。

2.3　元数据分类

数据仓库中的元数据涉及数据仓库设计以及管理的方方面面, 对元数据概念内涵进行深入的剖析, 建立比较全面的元数据分类将有助于元数据管理系统的管理和使用, 有助于数据仓库的成功实施。目前元数据分类方法主要有如下几种: ①按照功能和使用用户可以分为 3 种[3], 即商业元数据、技术元数据和操作元数据; ②按照使用对象和应用范畴可以分为 2 种[1], 即技术元数据和商业元数据; ③从元数据获取和元数据存取的角度可以分为 2 种[6], 即前仓元数据和后仓元数据; ④广义元数据和狭义元数据; ⑤根据元数据用法角色可分为 3 种, 即实现时元数据、主动运行时元数据和被动运行时元数据; ⑥从空间的角度可以分为 5 种[9], 即活动元数据、位置元数据、实体元数据、人群元数据和时间元数据。

另外, 还有许多种分类方法, 如分为业务元数据、数据库元数据和应用元数据[11]; 从数据仓库使用和组织的角度分为管理元数据和用户元数据 2 类等。不同的分类方案对于不同规模的项目和不同类型的公司的作用显然是不一样的。总而言之, 对元数据进行分类是一件非常有意义的事情。这将有助于加深对数据仓库中元数据的认识, 更好地使用元数据。但是目前的元数据研究还不够深入, 这些分类都还没有给出元数据的严密定义, 其根本原因是缺乏对元数据概念内涵的深入研究。

2.4　元数据的标准化

随着数据仓库的广泛应用, 元数据的重要性越来越被人们所认同, 涌现出了许多基于元数据的数据库工具。包括数据提取和清除工具、装载工具和分析工具等。分散在不同工具中的元数据的数据格式、数据模型和使用方法都不相同, 对用户来说使用复杂而且维护十分困难, 解决这个问题的关键是实现元数据的标准化, 以便共享跨企业、跨工具、跨平台的元数据。目前比较有影响的数据仓库元数据标准是元数据联盟 MDC(Meta Data Coalition) 的开放信息模型 OIM (Open Information Model) 和对象管理组织 OMG (Object Management Group) 的公共仓库元模型 CWM(Common Warehouse Metamodel)。

因为 2000 年 9 月 MDC 并入了 OMG, 所以只介绍 OMG 的公共仓库元模型 CWM。

OMG 在 2000 年发布了公共仓库元模型 CWM (Common Warehouse Metamodel) 规范。其主要目的是在异构环境下, 帮助不同的数据仓库工具、平台和元数据知识库进行元数据交换。CWM 模型既包括元数据存储, 也包括元数据交换, 它是基于以下 3 个工业标准制定的。

(1) UML。它对 CWM 模型进行建模。

(2) MOF(元对象设施)。它是 OMG 元模型和元数据的存储标准, 提供在异构环境下对元数据知识库的访问接口。

(3) XMI(XML 元数据交换)。它可以使元数据以 XML 文件流的方式进行交换。

这 3 个标准是 OMG 元数据库体系结构的核心, UML 定义了表示模型和元模型的语法和语义; MOF 为构造模型和元模型提供了可扩展的框架, 并提供了存取元数据的程序接口; 利用 XMI 则可以将元数据转换为标准的 XML 数据流或文件的格式, 以便进行交换, 这大大增强了 CWM 的通用性。OMG 元数据知识库体系结构如图 1 所示。

CWM 为数据仓库和商业智能(BI) 工具之间共享元数据, 制定了一整套关于语法和语义的规范。它主要包含以下 4 个方面的规范。①CWM 元模型: 描述数据仓库系统的模型; ②CWM XML: CWM 元模型的 XML 表示; ③CWM DTD: DW/BI 共享元数据的交换格式; ④CWM IDL: DW/BI 共享元数据的应用程序访问接口(API)。

OMG 称 CWM 标准是元数据标准中的里程碑[12], CWM 的业内支持者目前包括 Oracle、IB-

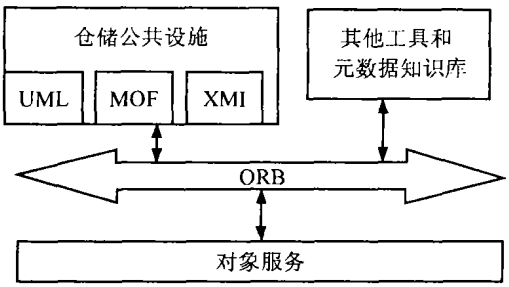

图 1 OMG 的元数据仓储体系结构

M、Unisys、NCR、Sun 和 Hyperion 等厂商, 它将在数据抽取、变换、交流、加载、集成和数据仓库分析领域提供一系列的 API、数据交换格式和其他类型的多种服务。

2000 年 12 月 22～23 日, 美国奥兰多, 6 个主要的 OMG 组织参加商 (IBM, Unisys, Hyperion, Oracle, SAS, Meta Integration) 成功运用 CWM 进行信息共享。这是 CWM 具有生命力的重要试验。而且 CWM 已经被 Java 社团着手扩展到 J2EE 体系结构当中, 形成 JMI (Java Metadata Interchange) 规范、用于 OLAP 分析的 JOLAP 规范和用于数据挖掘的 JDM API 规范。可以预见 OMG 组织的 CWM 标准将会成为数据仓库元数据领域事实上的标准, 在元数据管理系统的建立过程中应尽量参考这个标准。

3 元数据管理的策略

CWM 是 OMG 采纳的一个使用共享元数据的集成数据仓库和业务分析工具的开放行业标准。CWM 提供了基于模型的元数据集成体系结构所需要的, 用于描述问题域的语义完整的公共模型。如果构建数据仓库用到的各种软件产品、工具、应用软件和数据库产品能就 CWM 元模型达成一致, 它们就能理解 CMW 元模型的实例 (模型或者元数据), 因而可以很容易在各软件组件之间交换元数据 (共享的、重用的), 从前端的数据资源, 到转换和净化, 到终端用户分析, 再到数据仓库管理, 都能用 CWM 的元模型来建立。CWM 能够处理基于模型的元数据集成的面向模型的所有方面, 但是 CWM 并没有定义元数据管理的策略和元数据集成的体系结构。

要进行成功的元数据集成, 必须建立一个一致且合理的管理策略, 由这个管理策略为目标环境中的元数据集成、共享和重用制定目标和需求。因为无论元数据集成工具及其相应的软件产品功能多强或者多么健壮, 都不能代替一个合理、一致的元数据管理策略。

(1) 全局安全策略。由于元数据是一种具有高敏感性和战略价值的信息财富, 必须包含一个全面的安全策略以保证元数据能够得到充分保护;

(2) 对所有元数据源和目标的确定机制。通常情况下软件组件在不同时刻, 为不同目的承担不同的角色, 元数据管理策略必须明确定义什么组件在什么时候承担什么角色, 并且定义在数据仓库或者信息供应链的整个处理流程中与其他组件的协作;

(3) 对所有元数据元素的确认机制。这通常可以用某种惟一的标识符进行标记, 同时还要定义在不需要惟一标识的情况下如何处理的策略;

(4) 对每个元数据元素语义的一致理解。软件组件所用到的每一种元数据元素的语义必须存在一致, 这直接影响元数据的共享和重用;

(5) 每个元数据元素的所有权。必须确定哪些个体或哪些组件是一个特定元数据元素的最终所有者。要确保元数据的所有权最终属于元数据的主要项目相关人员 (数据仓库最终用户或者顾客), 而不是属于数据仓库的技术管理员或者开发者;

(6) 共享和修改元数据元素的规则。提供一种机制使得只有一个具有足够的权限的用户可以检出、修改和检入一个惟一的元数据元素;

(7) 重新发布元数据元素的规则。不管现有的执行这些操作的技术能力如何, 都必须为每一个受元数据变更影响的工具制定一个明确定义的策略;

(8) 元数据元素的版本控制。必须为被管理的元数据设立专门的版本控制规则;

(9) 元数据元素的重用目标。元数据管理策略应该通过整个元数据集成解决方案, 来为语义等效性和元数据元素重用的程度设立目标;

(10) 手工过程的消除机制和冗余元数据的消除机制。任何依赖人工干预的元数据集成解决方案对整个数据仓库的投资回报率都有一个动态的、负面的影响, 应该找出目前所有的手工过程并提出一个最终能够使得它们自动的计划。另外, 必须尽量消除元数据冗余以最大程度共享和重用元数据。

4 元数据体系结构

一个稳固的元数据体系结构必须具有集成性、可扩展性、健壮性、可定制性和开放性等特征。集成性, 即必须能够集成各种类型和来源的元数据, 并把得到的结果转换成有意义的、可访问的业

务元数据和技术元数据; 可扩展性, 即构造元数据仓储之后, 它能够进行扩展并随时间变化而改进, 否则这种仓储体系很快就会过时; 健壮性, 即必须有足够的功能和较高的性能来满足其所服务机构的要求, 必须能够同时支持业务用户和技术用户的元数据报表和视图, 并提供访问这些元数据的权限; 可定制性, 即如果公司使用元数据工具实现仓储体系结构, 就需要定制工具来满足元数据项目当前和今后的特定要求; 开放性, 即元数据集成和访问的所用的技术必须是开放灵活的, 否则避免不了对体系结构进行大规模的改动。以下简要介绍几个高级的元数据体系结构。

(1) 支持 CWM 的点到点的体系结构。如图 2 所示, 可以发现这种体系拓扑结构上类似于传统的集中式互连体系结构, 但是这种结构的中心连接点更多是概念上而不是物理上存在的, 却如同引入了一个物理上的逻辑中心点(即元数据中心存储库)一样, 引入逻辑上的逻辑中心点同样有益。从接口和元模型映射的角度看, 所有的工具都是通过一个逻辑上的 CWM 基础设施层集成的, 大箭头表示 CWM 适配器模块, 而不是与特定产品相关的元数据桥。例如 ODS 使用标准的 CWM 元数据接口为建模工具提供元数据, 建模工具通过同样的标准接口为其他各类元数据消费者提供元数据。所有这些元数据都与这个单一的、公共的 CWM 元模型保持一致。每个参与到体系结构的每一个工具只有一个公开的元模型(即 CWM 模型)。

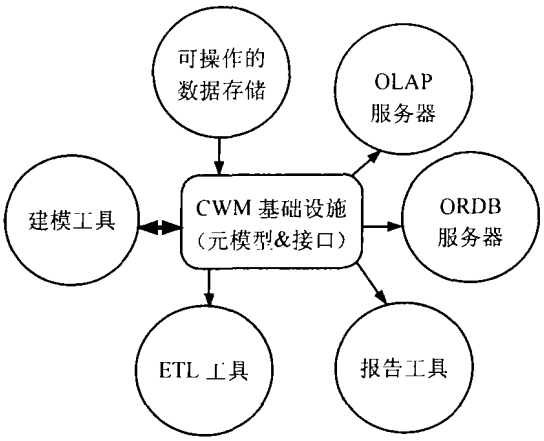

图 2　支持 CWM 的点到点体系结构

(2) 支持 CWM 的集中式体系结构。从图 3 可以看出物理集中的体系结构和逻辑上集中的体系结构之间几乎没有差别。事实上, 从基于一个公共元模型的标准元数据的接口的观点来看, 两者之间并没有本质的区别。只是元数据流的源和目标不同, 另外, 接口也可能存在差异。该体系结构要求用于定义和组织各种元数据的模式以及元数据本身都保存在全局的元数据仓储中。

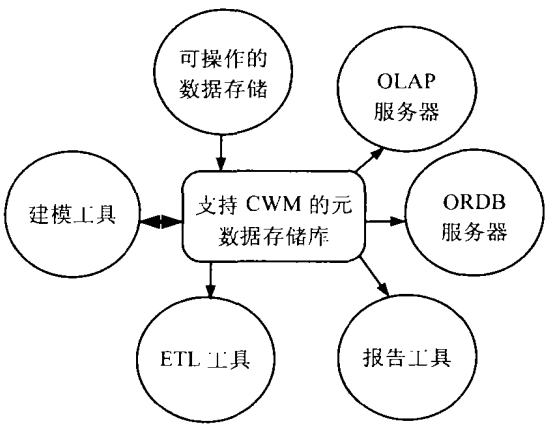

图 3　支持 CWM 的集中式体系结构

(3) 分散式的元数据体系结构。分散式元数据体系结构(如图 4) 的目标是创建统一和一致的元模型, 这种元模型要求定义和组织各种元数据的模式存储在全局元数据仓储和本地仓储的共享元数据元素中。不同仓储之间的元数据共享和重用的所有元数据必须先经过全局仓储进行处理, 但是共享和访问本地元数据则独立于全局元数据仓储。全局元数据仓储存储的是本地元数据仓储中的元数据的一个子集。这种体系结构适用于具有多种独立业务的公司。

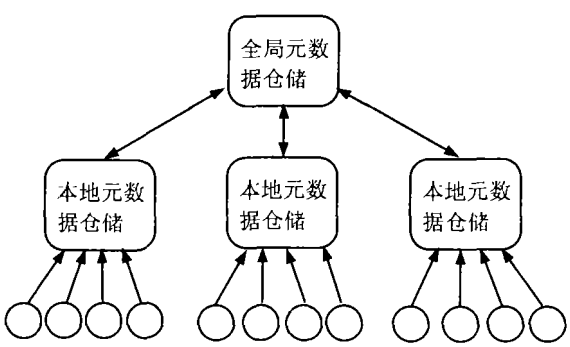

图 4　分散式元数据体系结构

(4) 双向元数据体系结构和闭环元数据体系结构。双向元数据体系结构类似于集中式元数据体系结构, 不同在于它允许工具之间共享元数据和允许企业在元数据仓储上进行全局性的改动并使整个组织感知这些变动。但是实现双向元数据面临着 3 个显而易见的挑战: ①要求元数据仓储必须含有将被反馈到元数据源的最新版本; ②需要系统化地跟踪和解决变动; ③需要构建几组额

外的处理接口,把元数据仓储连接回元数据源。

闭环元数据体系结构允许仓储把它的元数据反馈回公司的操作型系统,这里的概念与双向元数据体系结构有些相似,但元数据仓储仅仅将信息反馈给操作型系统而不是其他应用。闭环元数据体系结构提高了元数据项目设施的复杂性,要使从仓储反馈到操作型系统的元数据也能保存在该操作型系统中,元数据必须包含元数据的最新版本。另外,必须系统地跟踪元数据使用冲突,并构建程序接口将元数据仓储连接回操作型系统。

上述 2 种结构虽然使用的公司相对较少,但是这是元数据系统结构发展的大势所趋。

5 结 论

元数据的管理和维护是数据仓库实现过程中非常重要的一环,元数据就像一座桥梁,将数据仓库中的用户和使用者有机结合起来,它不仅在整个数据仓库系统,而且在决策支持系统中都起着非常重要的作用。

参考文献:

[1] 廖 王 ,王立刚,刘文煌. 构造数据仓库系统的元数据[J]. 计算机工程与应用, 2001(6):94-96.

[2] 戴超凡,刘青宝. 数据仓库中的元数据管理[J]. 计算机工程与科学, 2003, 25(4):54-57.

[3] 胡颖峰,卢美莲. 数据仓库中数据互通的研究[J]. 计算机应用研究, 2002(2):34-37.

[4] 戴超凡,邓 苏. 基于 GMM 的数据仓库管理与维护[J]. 国防科技大学学报, 2002, 24(6):81-86.

[5] 李道奇,马志军. 数据仓库中元数据的研究与应用[J]. 武汉理工大学学报, 2002, 24(7):76-78.

[6] 吕 波,王延章. 数据仓库元数据的界定与分类[J]. 信息与控制, 2001, 30(6):499-507.

[7] 曹蓟光,王申康. 元数据管理策略比较研究[J]. 计算机应用, 2001, 21(2):3-5.

[8] 戴超凡,邓 苏. 开放信息模型研究[J]. 计算机工程与应用, 2001(1):14-16.

[9] Sperley E. 企业数据仓库规划[M]. 陈 武,袁国忠译. 北京:人民邮电出版社, 2000.

[10] Inmon W H. 数据仓库[M]. 王志海,林友芳译. 北京:机械工业出版社, 2003.

[11] Dyche J. Data Warehouse Metadata and Middleware[J]. EAI Journal, 2000(9):71-76.

[12] 天极网. 对象管理协会采纳"元模型"标准[DB/OL]. http://it.sinohome.com/87016/88728.htm.

[13] Poole J, Chang D. 公共仓库元模型[M]. 彭 蓉,刘 进译. 北京:机械工业出版社, 2004.

Metadata Management in Data Warehouses

Wang Donglong, Chen Jian, Li Maoqing

Abstract: Metadata is the core of realization and management in data warehouse and also the difficulty of building a data warehouse system. The definition, function and classification of metadata are introduced. The standardization of metadata in data warehouse is discussed. The metadata management strategy and metadata architecture are probed.

Key words: data warehouse; metadata; OIM, CWM

Wang Donglong: Postgraduate; School of Computer Science and Technology, Xiamen University, Xiamen 361005, China.

[编辑:王志全]