第32卷 第19期 计 算 机 工 程 2006年10月
Vol.32 № 19 Computer Engineering October 2006
·软件技术与数据库· 文章编号：1000—3428(2006)19—0054—03 文献标识码：A 中图分类号：TP311

统一空间数据仓库权限管理分析与设计

李海波[1]，王丽珍[1]，杨 莉[2]，倪志凌[1]
(1. 云南大学信息学院，昆明 650091；2. 云南中医学院，昆明 650200)

摘 要：提出了一种在空间数据仓库系统下，基于主题和角色的统一的权限控制机制。将空间数据仓库的内部权限与外部权限、空间数据权限和属性数据权限统一起来。并用以支持国家自然科学基金资助重点项目——“三江并流”代表性植物空间数据仓库及其多样性研究项目的实验平台的构建。

关键词：空间数据仓库；数据仓库；规则；访问控制策略

Analysis and Design for Uniform Access Control Strategy in Spatial Data Warehouse

LI Haibo[1], WANG Lizhen[1], YANG Li[2], NI Zhiling[1]
(1. School of Information Science, Yunnan University, Kunming 650091; 2. Yunnan University of Traditional Chinese Medicine, Kunming 650200)

【Abstract】 An access control mechanism, which is uniform, topic-based and role-based, is proposed for spatial data warehouse. It unifies inner-privilege and outer-privilege, spatial data privilege and attribute data privilege of spatial data warehouse. It is used to support constructing for experiment platform in national natural science foundation project——the studying project about “the three parallel rivers area” representative plant spatial data warehouse and biological diversity.

【Key words】 Spatial data warehouse; Data warehouse; Rule; Access control strategy

从应用的角度而言，商业数据仓库兴起已为时不短，而空间数据仓库的出现，与GIS在军事、环境保护、旅游导航、城市规划等行业的广泛应用密不可分。空间数据仓库是在数据仓库基础上引入空间数据，根据主题从不同的GIS 应用系统中截取从瞬态到区段、从局部到全球系统的不同规模时空尺度上的信息，具有面向主题性与集成性、进行数据变换与增值等特性，为地学研究以及有关环境资源政策的制定提供最好的信息服务[1]。与空间数据库相比，空间数据仓库更加擅长于大量数据的集成计算分析和历史数据库相关的决策分析[2]；另外，空间数据仓库可为将来空间数据挖掘奠定良好的数据基础，因此，空间数据仓库已成为当今计算科学的又一技术热点。

由于数据仓库的最终用户身份的特殊性以及数据的价值，决定了数据仓库中的数据不可能为普通用户所用，工程项目中各行业中的数据仓库系统的安全性也一直受到重视，尤其是空间数据仓库，因此，在空间数据仓库构建初期，安全性便作为一个重要的任务摆在项目日程上。

1 存在的问题

当今主流的商用数据库产品，如Oracle、Microsoft SQL Server、Sybase等，所提供的数据仓库服务中都具备权限管理机制，如SQL Server2000 中的多维数据集角色[3]，使用户得以在多维数据集的不同粒度级别上定义安全性。然而，应用领域的需求变化多端，数据仓库构建完成后往往需要整合到外部的统一的信息管理应用系统中去展现，即数据仓库的权限管理要纳入外部应用系统的安全管理机制中并服务于不同的需求；到了空间数据仓库阶段，属性数据和空间数据的安全管理也需要统一。主要问题可列为以下几点：

(1)构建数据仓库的工具的权限控制与外部应用或管理系统的权限管理的统一问题。虽然数据仓库工具中也提供了比较完善的安全管理，但仅仅局限于数据库管理员或数据仓库构建者，因此不能满足最终用户的需要。

(2)针对业务功能的权限管理和面向主题的数据集合安全管理的统一问题，企业级的应用或管理系统的权限管理绝大部分针对业务功能，而数据仓库由面向主题的数据集合组成，问题在于如何使应用系统的权限管理也能对数据仓库的权限进行设置。

(3)属性数据的权限限制如何反映到空间数据的问题。数据仓库的数据聚合的结果一般需要用一幅地图或其它空间数据来展现，空间数据聚合如何再现非空间数据聚合的权限限制，这是一个极具挑战性的问题。

总之，对于空间数据仓库的建设，一个有效而统一的权限管理机制是十分值得付诸先行的。

2 分析与设计

在分析和总结以上问题并结合工程项目实践的基础上，本文提出设计的核心思想、数据结构和主要算法。用一个处理流程图来说明设计思想，如图 1 所示。

从图中 4 个阴影部分可以看到，权限管理贯穿整个空间数据仓库建设始终。核心思想体现在对 4 个阴影部分的权限管理的解释中。

基金项目：国家自然科学基金资助重点项目(60463004)
作者简介：李海波(1977－)，男，硕士，主研方向：空间数据仓库；王丽珍，教授；杨 莉，硕士；倪志凌，讲师
收稿日期：2005-10-26 **E-mail：**haierbopuhuixing@ynu.edu.cn

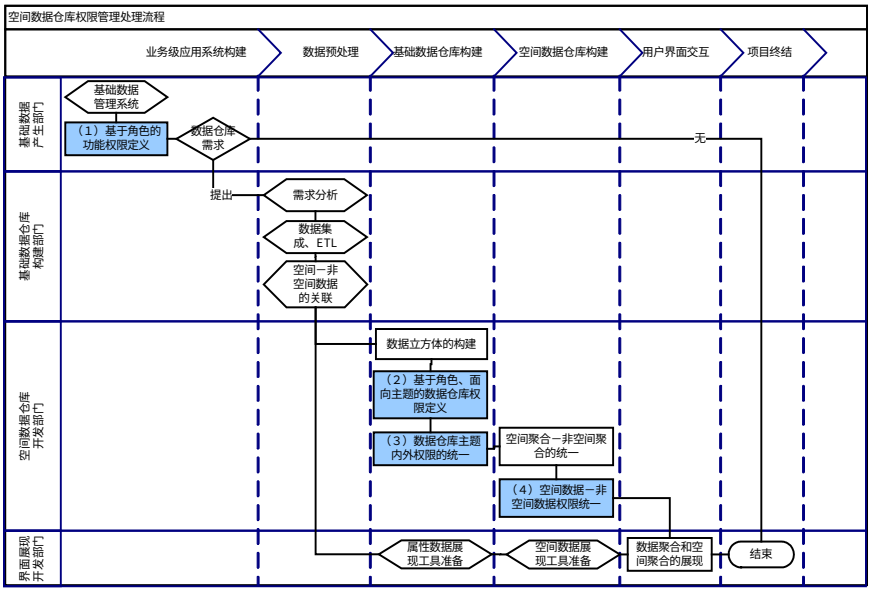


图 1 空间数据仓库权限管理处理流程

(1)基于角色的功能级权限管理定义。基于角色的功能集合授权机制是业务基础数据产生阶段的最主要权限管理。在此须定义：角色和用户是 m：n 的关系，角色和功能是 m:n 的关系以及部门和角色 1：n 的关系。图 2 所示的关系模型结构可以满足这一类授权需求。当用户以某一个角色登录系统时，该角色拥有的所有功能将载入系统，又当启动某一个功能时，与该功能有关的数据权限模式将约束该角色所能访问的数据，如根据数据权限模式中“数据过滤条件”属性的设置，只能查看自己生成的数据；另外，与该功能有关的操作权限模式将约束该角色所能应用的操作，如该角色只可以查询但不能使用保存功能。当完成这一部分的设计后，数据仓库部分的权限管理才能得以开展。

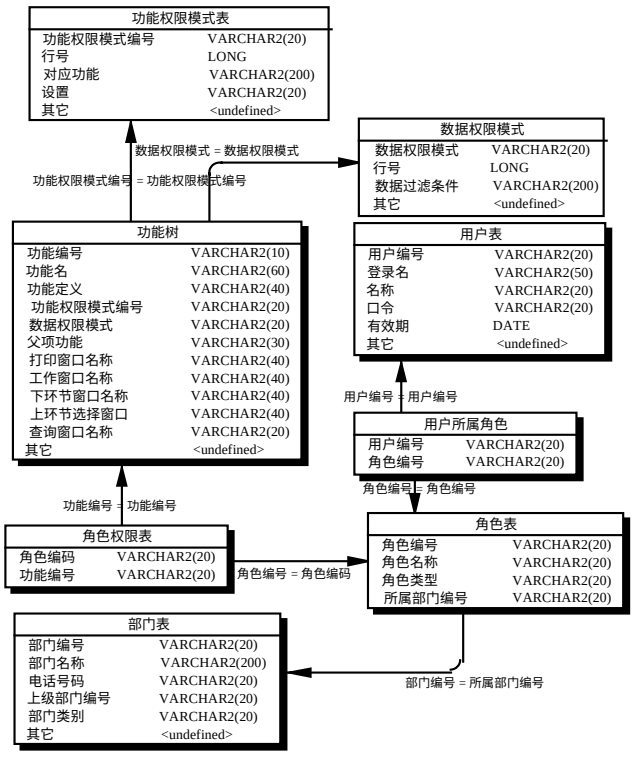


图 2 基于角色的功能授权数据结构

(2)基于角色、面向主题的数据仓库权限定义。角色与数据仓库主题也是 m：n 的关系。在此做了一个对等：将一个主题视为一个功能，从而使不同角色，根据其职能，可以分配到不同的数据仓库主题，这样处理的优点是可以兼容已有系统的权限管理。图 2 中的数据结构在满足面向主题的的权限设置不存在问题，然而，较业务基础数据而言，数据仓库中的数据更加敏感和重要，对于同一个多维数据集不同粒度数据的访问，则需要特殊的处理机制。这就是下面篇幅要处理的内容。

(3)数据仓库主题内外权限的统一。一般而言，数据预处理和数据立方体的生成主要借助主流商业数据库产品提供的工具完成，但数据的展现又往往需要嵌入到外部信息系统中，要保护数据仓库内对象和数据的安全，主要方法是使用多维数据集角色。为方便表达，以后篇幅使用内角色表示之。内角色使得开发人员得以在多维数据集的不同粒度级别上定义安全性。基于需要，可以在维度成员级别和(或)数据单元级别上保护多维数据集的数据的安全[3]。内角色一般在数据库系统中被定义，而其下用户可以是操作系统用户也可以是数据库自定义用户，这些区别都不是问题，问题是：外部系统的角色和用户，如何同内部数据库系统的角色和用户对应，由于两套系统的权限管理互相独立，因此这一问题相对复杂。设计思路体现在图 3 所示的系统处理流程中。

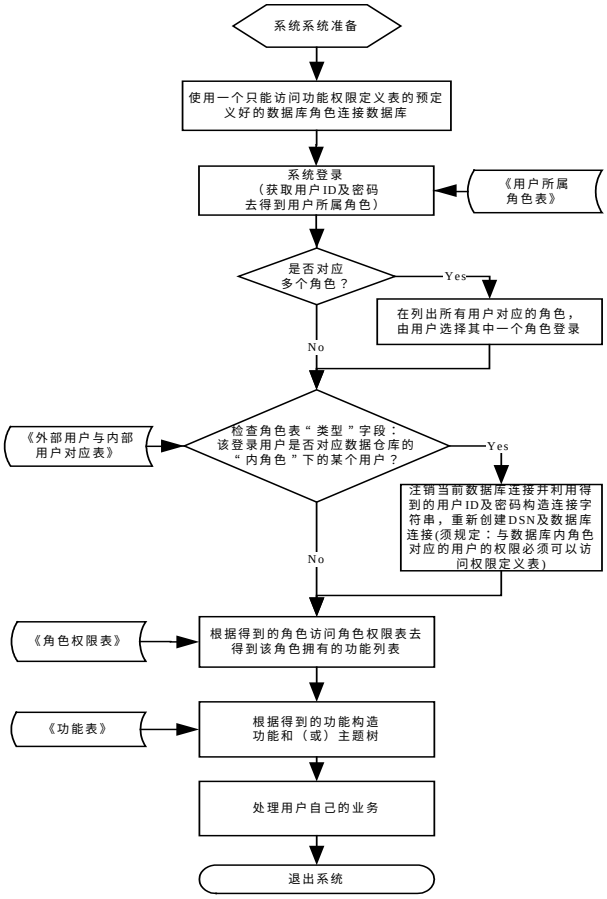


图 3 数据仓库主题内外权限的统一处理流程

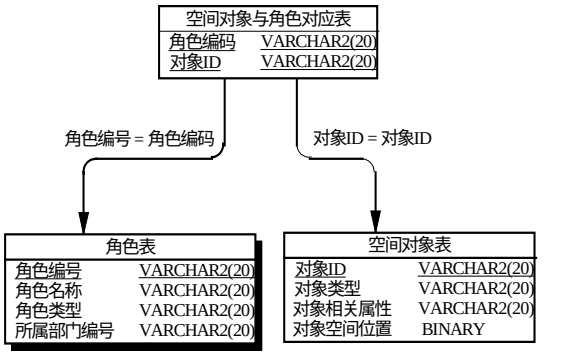
这一系统处理流程中需要进一步说明的是：实现外部系统某个角色下所属用户与数据仓库内部角色下所属用户的对应定义以及两阶段数据库的连接。这里需要做如下定义：

定义 1 每一个外部系统的用户被图 2 中的员工表的一条记录所代表。

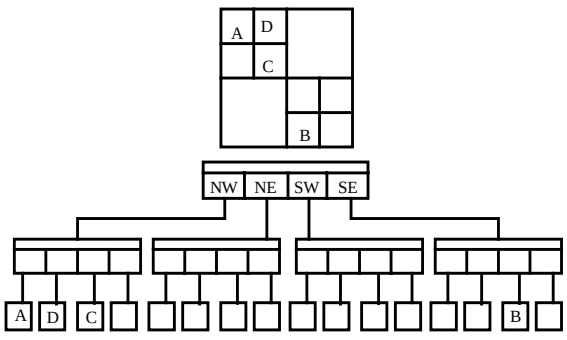
定义 2 如果有员工表的一条记录出现在图 2 中外部用户与内部用户对应表中，必须在数据库系统中定义一个与这条记录代表的员工的名称、密码相同的用户，并且这个用户从属于某一个内角色。

根据以上定义，所要做的只是对某些特殊用户专门定义其角色，使其不但拥有一般用户能访问的数据，也可访问有所限制的数据，如我们的项目中，对我国珍稀植物国家保护级别维度的访问是严格受限的。当然，有人会提出这样的问题：为何不为每个外部用户定义一个内部用户与之对应，这样可以简化处理流程。理论上是可以这样做，事实上，一个企业级数据管理系统的实施，将面对数以千计的用户，要为每一个用户作数据库的内部定义，工作量巨大不说，更重要的是维护企业员工岗位变换所带来的权限变更工作将持久而繁重。换一个角度讲，绝大部分用户使用的是一般的功能，基于功能级的权限管理已能满足需求；而高级用户的对企业的贡献和破坏可能是同一量级，因此针对少量这样的用户定义对应的内外权限是值得投入的。另外，系统中存在两阶段的数据库连接，这里使用了动态创建 ODBC 数据源功能，继而连接数据库，虽然性能有所下降，但换回灵活性和安全性。

(4)空间数据－非空间数据权限统一。这一部分具有很强的应用针对性，并不存在一套大一统的工具，二次开发在所难免。针对该项目，只有国家珍稀植物分布数据才是需要保护的数据，其它如植被、森林等分布属于一般数据，无需权限控制；珍稀植物的空间对象表示一般是具有半径的点对象，并且它们缺省是“ InVisible ”的。基于此提出相应数据结构和算法。



(a) 空间对象与角色之间的关系



(b)带权限控制的空间对象的数据组织结构

图 4 实现空间－非空间数据权限统一的数据结构

图 4(a)表示空间对象与角色是m:n的关系。为提高性能，在“ 对应表 ”中只存放需要高级角色与空间对象的关系。图 4(b)引入组织空间对象的树结构：其核心思想主要基于著名的MX_Tree[4]，是一种关于空间点对象的索引树结构。它把空间区域递归等分为西北、东北、西南、东南(NW,NE,SW,SE)4部分，直到每一个点都被一个叶节点所代表。因为项目中重要数据皆为点对象，所以使用这样的结构是合适的，并且结合项目定义了树节点结构如下：

```
typedef struct TNode{
    Object *pObject;//空间对象指针
  double Xval,Yval;//对象坐标位置
    CString PrLevel;
   /*存放权限模式等级：缺省为“ PUBLIC ”；
     若为“ AUTH ”，
     则需要检查“ 空间对象与角色对应表 ”中
     是否存在相应的记录*/
   metadatatype *pmetadata;
     //空间对象的相关元数据信息。
   struct TNode *NE, *NW, *SW, *SE;
  //分别指向地图东北、西北、西南、东南的指针。
}Tnode,
struct TNode *AQ_Tree;//定义树根节点。
```

以一个引例串联此部分设计：

例 1：统计“ 三江并流 ”地区国家以上级别的珍稀植物各自的总数，并以不同颜色表示其分布。

显示该例的处理过程如下：

```
long OnAnalysze(CString str_Role_no,CString str_PrMode)
                    //参数为：用户登录的角色号和用户的操作模式
{ TNode *THead
  Queue <*TNode> q_Object;
    //定义一个 TNode 指针队列。
  THead=q_Object.gethead()
  while(THead!=null)
  {          if(THead->NE==0)&&(THead->NW==0)
             &&(THead->SW==0)&&(THead->SE==0)
       {  SELECT COUNT(*) INTO n_count
                 from  空间对象与角色对应表
                 while (Object_ID=THead->pObject->ObjectID
                  and Role_code=str_Role_no
                       and Pr_mode=str_PrMode);
          if(n_count>0)||(THead->PrLevel=="PUBLIC")
          {  THead->pObject->visible=true;
                 THead->pObject->color=THead->pObject->
       getcolor(THead->pObject->type);
          }     }
       else
       {  q_Object.insert(THead->NE,THead->NW,
                                THead->SE,THead->SE);
       }     }
  return m_Map.refresh();//显示图形结果。
}
```

最后，空间数据仓库是一个综合复杂的系统，要设计一个完善的空间数据仓库，最重要的是搭建一个可伸缩的系统框架，不但可满足现在，还可用于将来。

3 结论

数据是空间数据仓库的最根本问题。本文提出的方法只是空间数据仓库的研究中众多问题的一步探索——数据访问控制策略研究。今后的研究中将涉及把权限控制对象扩展到任意形状的空间对象，空间对象索引结构的设计问题。

（下转第 59 页）