

基于 DBMS 的 MAPGIS7.0 版本管理的设计与实现

万 波^{1,2}, 周顺平^{1,2}, 陈 波^{2,3}, 方 芳¹

1. 中国地质大学信息工程学院, 湖北武汉 430074

2. 武汉中地数码科技有限公司, 湖北武汉 430074

3. 中国地质大学研究生院, 湖北武汉 430074

摘要: 提出了一种“版本化”的地理数据库管理模式和基于 DBMS(database management system)的版本管理模型, 讨论了其设计过程和层次结构, 给出了关键算法及 MAPGIS7.0 中版本管理的实现技术, 解决了传统 GIS 软件不能并发处理长事务的难题。该版本管理工具已经在国家“863”项目——地质调查空间数据库管理系统良好地运行, 能够满足多用户长期并发编辑地图的要求。

关键词: MAPGIS; 长事务; 版本。

中图分类号: P208; TP311

文章编号: 1000-2383(2006)05-0600-05

收稿日期: 2006-05-30

Design and Realization of MAPGIS7.0 Management Based on DBMS

WAN Bo^{1,2}, ZHOU Shun-ping^{1,2}, CHEN Bo^{2,3}, FANG Fang¹

1. Faculty of Information Engineering, China University of Geosciences, Wuhan 430074, China

2. Wuhan Zondy Cyber-Tech Co., Ltd., Wuhan 430074, China

3. Graduate School, China University of Geosciences, Wuhan 430074, China

Abstract: This paper presents a “versioned” managing mode to manage the geological database, and a model of version management based on DBMS. The design process and architecture are discussed, as well as the main arithmetic and technologies of version realization in MAPGIS 7.0. It solves the concurrency control problem of long-duration transaction which cannot be solved by traditional GIS software. Test shows that in actual application, users can do cooperative editing for a long term with MAPGIS7.0. Now it has been successfully applied to the national geological survey spatial database management system.

Key words: MAPGIS; long transaction; version.

0 引言

GIS 的许多应用都涉及长期的编辑制图工作, 如地质编图过程, 一次编辑可能历时数天, 甚至数月, 而且是许多地质工作者并发的编辑地理要素。这种长时间的要素编辑操作要求具有事务 ACID 特性, 即原子型、一致性、独立性、持久性。要素编辑的特点往往表现为下列需求: (1) 从开始编辑到结束编辑, 所有的编辑操作要么都取消、要么都提交, 无论取消或提交, 都不破坏地理要素的一致性和完整性;

(2) 所有的编辑结果只有编辑人员可见, 在提交编辑之前, 所有编辑结果都能够保存, 但对其他人不可见; (3) 多人同时编辑, 相互之间不受影响; DBMS 提供的事务, 一般都采用加锁机制。在事务处理过程中, DBMS(database management system) 将锁定相应表的相应行记录, 直到事务结束。这就要求用户的事务时间尽可能的短, 并且可能产生死锁情况。因此 DBMS 的事务不能满足 GIS 编辑过程中的这种长时间并发访问的要求 (Michael, 2003)。

因而设计一个版本管理工具, 能够以版本的方

基金项目: 国家“863”计划: 基于大型 GIS 地质调查空间数据库管理系统 (No. 2003AA132010); 中国地质大学 (武汉) 青年教师基金资助计划项目 (CUGQNL0532); 武汉市重大科技产业化专项项目。

作者简介: 万波 (1975—), 男, 讲师, 博士研究生, 主要从事 GIS 理论与软件开发、空间数据库方向研究。E-mail: magicwan1105@163.com

式对地理数据库进行管理,能够处理 GIS 长事务,并满足 GIS 中长时间协同编辑的应用需求,具有重要的现实意义.本文介绍了 MAPGIS7.0 版本管理的设计思想、物理设计、功能模块以及关键实现算法.

1 版本管理原理

基于 MAPGIS7.0 平台的长事务实现机制有基于状态和版本这 2 个概念,通过版本控制,让多个用户可以直接编辑某个地理数据库而不用明确地锁定要素或者复制数据.

状态是地理数据库变化过程中某一瞬间的标识.任何改变地理数据库的操作都产生新状态.地理数据库中的这些状态可以组织成一棵树,在这棵树的线性结构中描述了各个状态的父子关系;版本是一个命名的状态.地理数据库中的每一个版本都明确指向一个具体的状态.不同版本在数据逻辑上相互区分.这样,我们在进行要素编辑时,可以为同时进行分工合作的人员定义各自的版本,每个人都在自己的版本空间下工作,不受其他人编辑的干扰,完成编辑后,再进行版本的合并.

版本管理模式没有使用 DBMS 对一般事务采用加锁处理并发的方式,多用户的并发是建立在各自的状态分支上的.用户在自己的状态分支上所作的任何操作完全由该用户自己控制,不受其他任何

用户的影响.这种模式实现了事务的乐观并发性.这意味着当开始一个长事务时,没有任何锁加到要素上.在这种模式下允许引入编辑冲突,当提交事务时,检测冲突并协调解决冲突.

乐观并发性是适于 GIS 应用的,因为相对于地理数据库大小来说,要素编辑的数量是小的.在实际的地图编制过程中,编辑冲突并不经常发生,并且比起长事务期间锁住要素来说,协调冲突的代价比较微小(廖国琼,2000).

2 基于 DBMS 的结构设计

版本管理的主要逻辑思想通过记录数据库的状态来维护地理数据库的版本,在 DBMS 中可以通过一组特定模式的关系表格在整个空间数据库和要素类 2 个层次上来实现.

在数据库层次上,基于 MAPGIS7.0 SDE 设计版本管理数据字典如下:MPDB_STATES,MPDB_STATE_LINEAGES,MPDB_VERSIONS,MPDB_MVXCLS_MODIFIED.其中 MPDB_STATES 表用来存储地理数据库变化过程中产生的状态,每个状态有它对应的状态分支,MPDB_VERSIONS 表存储地理数据库中版本的信息,其中每个版本对应一个地理数据库的状态,MPDB_STATE_LINEAGES 表存储地理数据库中每个状态分支中存在的状态列表,用来提高系统查询的效率;同时还提供

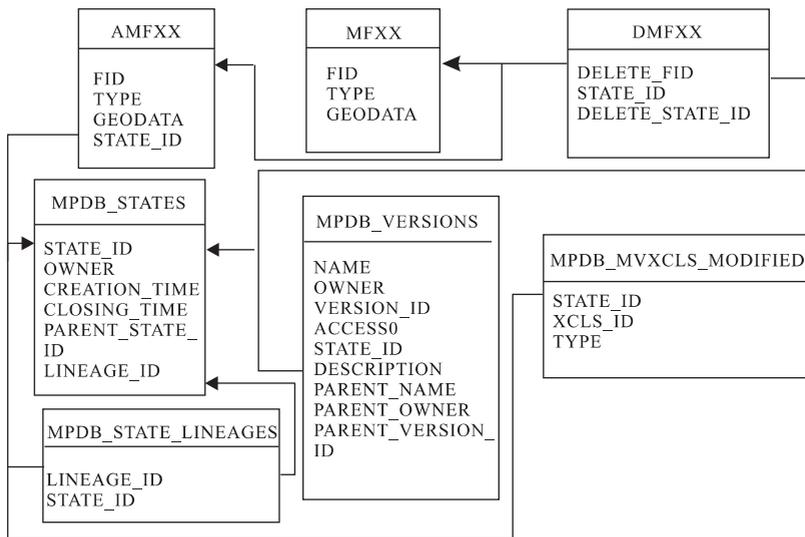


图 1 版本管理关系结构

Fig. 1 Relation of version management

MPDB_MVXCLS_MODIFIED 表存储状态改变类的信息,每个类在哪个状态下被修改,在该数据字典中都有记录,该字典在版本协调过程中进行冲突检测时使用。这 4 张表对于整个地理数据库来说是全局共享的(任娟,2005)。

在要素类层次上,基于 MAPGIS7.0 SDE 设计的 AMF、MF、DMF 3 个表,来支持对要素类进行版本管理。其中 AMF 表存储添加的要素信息,AMF 表中的 state_ID 字段表示该添加动作对应的地理数据库的状态;DMF 表存储删除的要素信息,其中 deleted_state_ID 字段表示该删除动作对应的地理数据库的状态。MF 表存储原始状态对应的数据。特定版本下的数据检索可以通过对 MF 表+AMF 表-DMF 表组合查询来实现。

在空间数据库层上的 4 个数据字典表和要素类层次上支持版本管理的类的 AMF 表/DMF 表联合起来即可完成对要素类的版本管理操作。图 1 给出了用于实现地理数据库版本管理的数据库模式的关系结构,该图虽然是地理数据库版本管理的实现模式,同样也是建立在关系数据库基础上的面向对象地理数据库模型版本管理实现的一般方法。

3 功能设计

在实际地理数据生产中,版本管理模块不仅要具有地图编辑的所有功能,还需要提供有效的机制来对不同用户、不同时间的各自编辑结果进行浏览、比较和取舍;综合实际地理数据实际生产和面向实体的地理数据管理的需要;MAPGIS7.0 版本管理应具有版本创建、删除、归并、冲突解决等功能和机制(吴信才,2003)(图 2)。

(1)创建版本。地理数据库创建的时候将创建一

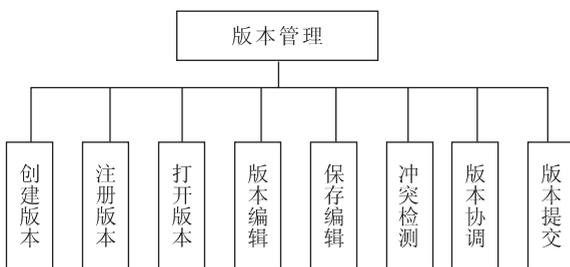


图 2 版本管理功能模块

Fig. 2 Function module of version management

个“缺省”版本,它是以后创建任何版本的父版本或者祖先版本。用户可以根据需要选取地理数据库已经存在的任何版本作父版本来创建版本,并且确定版本的访问权限。版本权限包括私有的、保护的和公有的。

(2)注册和注销版本。系统为每个支持版本管理的类提供了将该类支持和取消版本管理能力的方法:注册版本、注销版本。注册版本后,该类才具有长事务处理的能力,才能够进行版本的编辑,协调和提交等操作;而注销版本则使该类就失去了版本管理的能力,不能进行版本管理的相关操作。

(3)打开版本。用户打开要素类的时候可以指定打开哪个版本。要素类等在该版本下打开后,用户对类的编辑(如 Append, Update, Delete 等)将只在该版本下可见。

(4)版本编辑。版本化的类在开始编辑后,才能进行要素、几何或空间点、线及属性表格等进行添加、删除、更新等操作;不开始编辑将不能进行正常的编辑操作或操作结果不能保存。

(5)保存编辑。保存编辑将使对要素类的编辑结果得以保存,不保存的结果将视为放弃开始编辑后的所有操作。

(6)冲突检测。保存编辑结果时,系统将在下列 2 种情况下检测冲突:①系统将检测是否有多个用户打开并编辑同一个版本,并对同一个要素进行了编辑,若有将判定是否有冲突。②系统将检测是否有多个用户分别打开不同的版本,并对同一个要素或对象进行了编辑;这些编辑结果各自保存时并不存在冲突,但合并版本时,则将判定是否存在冲突。冲突根据对同一要素或者实体进行的编辑类型的不同分为:更新—更新冲突,更新—删除冲突和删除—更新冲突。

(7)版本协调。在版本下编辑的结果只能向父版本协调。协调的结果就是将没有冲突的动作合并,将存在冲突的操作选出来,交给用户仲裁。系统无法确定那种编辑更符合用户的最终要求时,就将 2 个版本下的不同编辑判定为冲突。仲裁冲突的用户根据需要主观地决定采用那个版本下的编辑,或者 2 个都不采取,而选择退回编辑之前的状态,以便采用另外用户认为更合理的处理方式编辑该要素或者实体。冲突集中时可以单个解决,也可以同时解决。

(8)版本提交。用户在解决完冲突后,将结果提交到当前版本或父版本。提交后产生冲突的 2 个版本的编辑结果将变为一致;不提交则冲突的解决结

果将不能保存。

4 实现模型

4.1 要素编辑实现模型

要素的编辑过程可以分解为添加要素、更新要素、删除要素 3 个基本的动作。添加一个要素, AMFXX 表中添加一条记录, 记录新添加要素的信息(表 1); 更新一个要素的动作可以分解为删除原来的要素, 添加一个新的要素, DMFXX 表中添加一条记录(表 2), 记录删除该要素的信息, 然后在 AMFXX 表中添加一条记录(表 3), 记录新添加要素的信息, 在此过程中, 新的要素 ID 保持不变; 删除一个要素, DMFXX 表中添加一条记录(表 4), 记录删除该要素的信息。

4.2 要素范围检索实现算法

矩形范围检索要素的实现算法如下:

(1)从线索树 MPDB_STATE_LINEAGES 表中找到指定状态的线索分支上的祖先状态集合 CS;

(2)在 MF 表中查询出所有满足给定矩形范围

表 1 添加要素:向 AMFXX 表中添加一条记录

Table 1 Add a feature: insert to AMFXX

FID	GEODATA	STATE_ID
1	1,1……	1

表 2 更新要素:向 DMFXX 表中添加一条记录

Table 2 Update a feature: insert to DMFXX

DELETE_FID	STATE_ID	DELETE_STATE_ID
1	1	2

表 3 更新要素:向 AMFXX 表中添加一条记录

Table 3 Update a feature: insert to AMFXX

FID	GEODATA	STATE_ID
1	1,1……	1
1	1,2	2

表 4 删除要素:向 DMFXX 表中添加一条记录

Table 4 Delete a feature: insert to DMFXX

DELETE_FID	STATE_ID	DELETE_STATE_ID
1	1	2
1	2	3

条件的要素集合 CF;

(3)在 DMF 表中查询所有 CS 集合包含的状态中删除了 0 状态的要素集合 CD1;

(4)在 AMF 表中查询出满足给定矩形范围条件的要素集合 CA;

(5)在 DMF 表中查询出所有 CS 集合包含的状态中删除了 CS 状态中添加要素的集合 CD2;

(6)最后要素检索结果可以表示为 $(CF - CD1) \cup (CA - CD2)$ 。

例如在要素类 283 中查找与矩形范围 ($x_{min} = 100, y_{min} = 200; x_{max} = 300, y_{max} = 400$) 相交, 且状态为 199, 线索分支为 197 的要素集合, 其 SQL 实现模型如下:

```
SELECT bf.FID, 0 STATE_ID, bf.TYPE,
bf.INFID, bf.DATALEN, bf.DATA
FROM YANG.MF283 bf WHERE (bf.FID,
bf.FID)
```

```
NOT IN (SELECT DELETE_FID, DELETE
_FID FROM YANG.DMF283 WHERE DELETED
_STATE_ID IN
```

```
(SELECT l.STATE_ID FROM YANG.MP
DB_STATE_LINEAGES l
```

```
WHERE l.LINEAGE_ID = 197 AND l.
STATE_ID <= 199))
```

```
AND bf.XMIN <= 300 AND bf.XMAX >= 100
AND bf.YMIN <= 400 AND bf.YMAX >= 200
```

UNION ALL

```
SELECT af.FID, af.STATE_ID, af.TYPE,
af.INFID, af.DATALEN, af.DATA
```

```
FROM YANG.AMF283 af WHERE (af.
FID, af.STATE_ID)
```

```
NOT IN (SELECT DELETE_FID, STATE_
ID FROM YANG.DMF283 WHERE DELETED_
_STATE_ID IN
```

```
(SELECT l.STATE_ID FROM YANG.MP
DB_STATE_LINEAGES l
```

```
WHERE l.LINEAGE_ID = 197 AND l.
STATE_ID <= 199))
```

```
AND af.STATE_ID IN (SELECT l.STATE_
ID FROM YANG.MPDB_STATE_LINEAGES l
```

```
WHERE l.LINEAGE_ID = 197 AND l.
STATE_ID <= 199)
```

```
AND af.XMIN <= 300 AND af.XMAX >= 100
```

AND af. YMIN \leq 400 AND af. YMAX \geq 200.

4.3 要素冲突检测算法

冲突的产生都是在父子版本所处的不同的状态分支对同一个要素进行了更新或者删除操作产生的. 更新—更新冲突的检测算法如下:

- (1) 查询出 2 个状态分支的最大公共状态 S_c ;
- (2) 查询出父版本在 S_c 状态到状态 S_t 上所有更新的要素的集合 CU_t ;
- (3) 查询出子版本在 S_c 状态到状态 S_s 上所有更新的要素的集合 CU_s ;
- (4) 更新—更新冲突的结果可以表示为 $CU_t \cap CU_s$.

更新—删除冲突的检测算法如下:

- (1) 查询出 2 个状态分支的最大公共状态 S_c ;
- (2) 查询出父版本在 S_c 状态到状态 S_t 上所有更新的要素的集合 CU_t ;
- (3) 查询出子版本在 S_c 状态到状态 S_s 上所有被删除的要素的集合 CD_s ;
- (4) 更新—更新冲突的结果可以表示为 $CU_t \cap CD_s$.

4.4 要素状态压缩算法

定期压缩地理数据库中的冗余状态可以有效的提高地理数据库检索要素的性能, 具体的算法如下:

- (1) 遍历状态树, 决定哪些状态是冗余状态, 可以被压缩; 没有版本指向该状态节点时只有一个子状态节点的状态可以被压缩;
- (2) 对能够压缩的状态, 需要消除该状态在版本管理相关的数据字典(MPDB_STATES 表、MPDB_STATE_LINEAGES 表、MPDB_MVXCLS_MODIFIED 表)和相关 AMF 表、DMF 表中的影响.

5 结论

目前基于大型 GIS 的版本管理模型已经应用到国家基于大型 GIS 地质调查空间数据库管理系统中. 经过试验检测, 版本化后的性能已经接近没有注册版本的性能, 说明本文设计的版本管理模型及实现思路确实可行, 并能够很好地满足实际应用的需求.

References

- Liao, G. Q., 2000. An optimistic concurrency control method for supporting engineering design transaction. *Computer Engineering*, 26(7): 24—25 (in Chinese with English abstract).
- Michael, Z., 1999. Modeling our world. Environmental Systems Research Institute, Inc., California.
- Ren, J., 2005. Principles of geodatabase version control. *Land and Resources Information*, 6(3): 41—45 (in Chinese with English abstract).
- Wu, X. C., 2003. MAPGIS7.0 white paper. Wuhan Zondy Cyber-Tech Co., Ltd., Wuhan (in Chinese).

附中文参考文献

- 廖国琼, 2000. 一种支持工程设计事务的乐观并发控制方法. *计算机工程*, 26(7): 24—25.
- 任娟, 2005. Geodatabase 版本控制原理剖析. *国土资源信息化*, 6(3): 41—45.
- 吴信才, 2003. MAPGIS 7.0 技术白皮书. 武汉: 中地数码科技有限公司.