

支持协同设计的工程数据库版本管理^{*}

王潜平 林宗楷 郭玉钗

(中国科学院计算技术研究所 CAD 开放实验室 北京 100080)

摘要 计算机支持协同设计是计算机支持协同工作 CSCW (computer supported cooperative work) 的一个应用领域,也是 CAD 技术的进一步发展. 本文论述了计算机支持协同设计的基本概念、内容. 讨论了用于计算机支持协同设计的工程数据库版本管理的思想、方法,并就多库分布式工程数据库的版本管理及其实现进行了探讨.

关键词 版本管理,计算机支持协同工作,工程数据库,CAD,分布式.

目前大多数 CAD 软件仅支持单个设计者的独自设计和人机交互,即使支持多用户的系统也很少考虑设计者之间的交互. 在开发新产品或工程项目中,设计阶段不仅是由众多的设计者分工协作,还应听取制造或施工人员、管理人员和用户的意见,才能较完满地完成设计工作. 因此必须研究支持群体协同设计^[1]的 CAD/CAM 系统. 基于分布式工程数据库的协同设计方案,就是这样一种计算机支持协同设计的环境. 它对现有的 CAD 系统进行改造和扩充. 本文介绍了计算机支持协同设计以及用于协同设计的工程数据库管理系统 EDBMS,讨论了协同设计中版本管理的几个问题,最后介绍了用于协同设计的多库分布式工程数据库的版本管理的模型及其实现.

1 计算机支持协同设计

它是指支持有关人员集体进行设计工作的一种环境,是以群体工作目标为核心,组织有关人员进行协同工作. 它强调设计者采用群体工作方式,以发挥各设计者在各方面的作用、才干等主观能动性因素. 传统设计方法所造成的项目管理与设计之间、不同设计阶段之间、设计工作与生产制造之间的脱节,更由于不能及时听到用户的意见,因此容易导致设计周期过长、设计费用高、设计质量不易保证等弊病,这些问题将随着计算机支持协同设计的出现而得到改善和克服. 协同设计中参与设计的不仅有设计人员,还有管理人员、现场工程技术人员直至用户. 这些参与者可能处在不同的地理位置上,还可能使用不同的软硬件系统,但通过各种网络将他们联系起来,使他们能同步或异步地进行设计和讨论,通过讨论和交流不断

^{*} 本文研究得到国家“八五”攻关项目资助. 作者王潜平,1964年生,讲师,主要研究领域为工程数据库、CSCW、MIS. 林宗楷,1934年生,研究员,博士生导师,主要研究领域为工程数据库、CAD与CSCW. 郭玉钗,女,1936年生,研究员,博士生导师,主要研究领域为工程数据库、CAD与CSCW.

本文通讯联系人:王潜平,北京 100080,中国科学院计算技术研究所 CAD 开放实验室

本文 1995-10-16 收到修改稿

缩小矛盾和分歧,取得协调一致.协同设计使不同部门的人员共同参与项目的设计开发过程,使他们密切合作,加快设计速度,降低设计费用,提高设计质量和效率.

协同设计中有效的数据管理变得日益重要,一个能支持协同设计的具有处理工程数据特点的工程数据库系统是必要的.分布式工程数据库系统是建立集成化协同式 CAD/CAM 系统的核心技术之一.通过它实现设计、分析、绘图、加工等各个阶段的信息共享;及时协调发现的矛盾与冲突,达到各类人员之间的协同设计;提高集成化 CAD/CAM 系统的运行效率.

2 工程数据库管理系统—EDBMS

工程数据库管理系统—EDBMS^[2]是我室研制开发的一个分布式工程数据库管理系统.它可用于集成化电子、机械、建筑等 CAD/CAM 系统中.系统采用关系与网状混合的数据模型,并吸收面向对象的设计思想.为了适应工程设计领域中数据管理的特点,EDBMS 在数据库系统基本属性的基础上扩充了一些属性如:V 型属性,是一种变长属性,通过 V 型属性支持变长数据的表示和处理;L 型属性,是一种长记录类型,它支持工程长记录的表示和处理.通过这些特殊属性从而具有对格式化数据和非格式化数据(包括文本、图形、图象、声音和动画数据等多媒体数据)统一管理的能力.

系统除了具备传统数据库管理系统的基本功能外,还具备管理工程数据的一些特殊功能,如描述工程应用中复杂对象之间关系(表与表、表与记录、记录与记录等关系)的能力;支持工程长事务处理;数据模式的动态修改和扩充;提供了客户/服务器环境下的分布式管理,它为协同设计提供了支持;此外开发了重要的版本管理功能.

3 协同设计中版本管理的几个问题

协同设计是由多个设计者参加的,并在网络环境下进行的.这使得用于协同设计的工程数据库的版本管理变得复杂.

3.1 保留设计的历史

协同设计中每个设计者背景不同,对同一个设计对象,会根据各自的知识、经验提出不同的设计方案,产生不同的设计版本.一个设计常常要对多种设计方案(或设计版本)进行分析、比较、优化、综合,最终才能完成.协同设计就是不断综合各设计者的设计和意见、不断地修改完善,因此不仅要保留最终设计结果,设计过程及设计过程中有关信息和方法也要被保留.从而要求版本管理能够保留和管理设计的历史、不同的设计方案和动态变化的设计模式等.^[3]

3.2 信息量大且分布于整个网络

协同设计中设计信息和设计方法分布于网络服务器和客户机上.一些共用信息和方法存放在服务器上,为所有设计者共享.而设计者的私有设计信息一般存放在自己的客户机上.为了协同,一般还要将部分设计信息对其他设计者开放,供其他设计者共享,这样信息就分散在整个网络上.同时为了协同,设计者之间会利用各种媒体进行交流,而具有声、文、图等的多媒体交流,会产生大量的信息.这就要求系统能处理和存储多媒体信息,并使信息冗

余小,存取速度尽可能快.信息量大且分散是版本管理所要解决的主要问题之一.

3.3 设计信息的多库存储

为了有效地管理分布于整个网络的设计信息,在我们的系统中将它们组织成多个数据库来进行管理.如将一些基本信息组织成系统数据库存放在服务器上,供设计者们共享.而设计者的设计信息形成私有库存放在客户机上.这样划分成几种数据库后可以减少版本信息的存储,因为系统信息变化较小,版本的演绎也少.划分成多库后,每个库相对较小,从而对于合并、删除等版本操作来讲,要处理的信息量也较少,处理也容易一些.并且对单个库可以用较为成熟的单库管理技术进行处理.从而使得系统的存储冗余小,管理层次清楚,便于工程应用.

3.4 协同设计中多级的版本管理

协同设计中要管理的信息很多,为了管理好这些信息,我们将用于协同设计的多库分布式工程数据库系统的版本管理分成以下几个主要层次:①对设计者在客户机上所进行的私有设计的版本管理以及系统共享信息本身的版本管理,这是最基本的版本管理.②对设计者调用系统信息及其他协同设计者的信息所进行的项目子设计的版本管理.③要对整个项目的设计进行版本管理.这样采用分级版本管理的措施,降低了管理的难度,并可以在应用中逐步提高完善.对于设计者私有设计版本的管理采用有向无循环图的管理方法,对此另有文章讨论,不再赘述.整个项目的版本管理涉及到项目管理、设计者间的交互通讯等,亦将另文讨论.下面就设计者的项目子设计的版本管理进行讨论.

4 协同设计中版本管理模型

4.1 版本管理模型

根据协同设计中版本管理的特点以及多库分布式工程数据库管理的实际情况,吸取面向对象的基本思想,结合整体一部分结构原则,采用了多库多级版本管理的策略.系统将设计信息分成多库后,每个库可视为一个广义的类,从而可以把各个库的信息进行封装,如将系统信息封装在系统信息版本中,把设计者的私有设计信息封装在设计者的私有设计版本中,这样把相关的内容结合在一起,减少了各部分之间的交叉,保证了设计信息的一致性,并且把变化程度不同的数据库版本分离开来,从而可以有重点的进行版本管理.在目前情况下,处理的重点放在设计者版本管理上.项目子设计是在私有设计的基础上结合其他设计版本而形成的,对此我们采用基于多继承的多源有向无循环图的管理模型,使其共同属性更易显见,并且可以提高信息的共享程度.具体在项目子设计版本管理中,就是在基本的版本管理基础上,扩充了一些数据结构,并以此为基础来进行管理.在我们的模型中设置了系统版本字典以及设计者版本字典等数据结构.系统版本字典是一个共享资源,存放有目前系统的版本变化情况,它存储在服务器上,为所有设计者共享.设计者要产生新的设计版本必须先申请系统版本字典,在系统版本字典的统一控制下产生新版本.为保证整个系统版本的有序发展,系统版本字典总是保持当前系统最高版本序号;设计者每产生一个新版本,系统版本号就加1.在客户工作站上也要将系统版本信息和有关设计版本信息写入设计者版本字典中.通过这些版本字典,使设计者之间可互相调用有关的设计版本,调用时先根据自己的版本字典中的系统版本号去查找其他设计者对应的设计版本.有时某设计者可能没有与系统

版本号直接对应的版本,这时取其最接近的设计版本.

图 1 是 3 个设计者的设计版本和一个系统共享信息版本的版本演变图. 对于设计者 B 来讲,它的设计版本 B3 对应于系统版本号 V7,若它的一个项目子设计需包含设计者 A、设计者 C 及共享信息,则其对应的项目子设计版本包括:A2,B3,C1,D1 这样几个基本版本. 同样其 B5 版本对应于系统版本号 V15,它的项目子设计版本包括:A3,B5,C5,D2 这样几个基本版本.

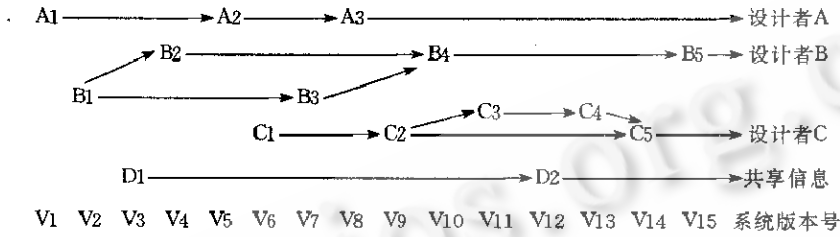


图1

对于版本字典的操作,共享信息版本字典的修改必须有授权才可进行. 设计者版本字典的修改一般由设计者本人进行,而系统版本字典的操作在设计者申请后由系统进行. 这样虽对系统的并发性有所限制,但实现上较为简洁,而且基本上也能满足工程需要.

为处理多媒体信息,采用了 V,L 属性的方法,在其存储上尽量采用压缩存储.

为了减少信息存储量,根据实际工程设计中前后设计版本间的变化一般不是太大这一情况,在版本数据的存储上,采用了向前版本管理策略存储版本的差集,即从一个具有完整数据的版本(源版本)开始,在版本的衍生发展过程中,只存与其直接前驱版本间的差,并且采用以记录为基本单位存储. 即后继版本中只保存更新了的数据. 采取存储版本差集的方法,可以节省大量的存储空间.

4.2 版本管理数据结构

(1) 系统版本字典

它包括:系统版本号、版本的设计者名(用户名)以及对应的设计者版本名等.

```

struct{
    INT32 Version-No;
    char user-name[MAX_LENGTH];
    char user-version-name[MAX_LENGTH];
}ESsys-dict
  
```

(2) 设计者版本字典

各设计者在客户工作站上分别设置一个自己的版本字典用于记录自己的私有设计的版本变化情况. 它的基本内容包括:设计者名(用户名)、版本名、对应的系统版本号、版本的状态(源版本、一般版本、属于项目子设计版本)、前驱版本名、后继版本名以及该版本中所包含的表等.

```

struct{
    char user-name[MAX_LENGTH]
    char version-name[MAX_LENGTH];
    INT32 Version-No;
  
```

```

char prior-version-name[MAX_LENGTH];
char succ-version-name[MAX_LENGTH];
char version-state;
EStab-name table-names[MAX_LENGTH];
}ESver-dict

```

在多库分布式情况下,设计者的版本中所包含的表可能来自于设计者本身的设计版本,也可能来自于其他设计者的设计版本以及系统共享信息的某一版本,因此又设计了管理表信息的数据结构.它主要包括:表名、表所属的数据库名、所属的版本及所属的设计者等内容.

```

struct{
    char table-name[MAX_LENGTH];
    char Database-name[MAX_LENGTH];
    char Version-name[MAX_LENGTH];
    char Version-owner[MAX_LENGTH];
}EStab-name

```

(3) 版本差集索引

在存储上采用了存版本差集的形式.记录版本信息变化情况的机构,即差集和差集索引.对各个版本均设置有差集索引.其结构是:

```

struct{
    INT32 dbk;
    char operator;
}EStab-dml

```

其中 dbk 是数据库主码,它的值在整个系统中是唯一的,并且唯一对应于一条数据库记录. operator 是在 dbk 所指的记录上进行的操作,如对记录的删除、修改、增加等.

5 版本管理功能及算法

目前提供的版本管理功能有:置当前版本、版本的合并、版本的删除等.

5.1 置当前版本

置当前版本也就是要把某设计者的项目子设计版本设置为当前要操作和处理的版本.首先要找到该版本,接着要检查是否属于项目子设计的版本,如是项目子设计的版本,则根据设计者版本字典中的版本号找到系统版本号,再根据系统版本号找到所使用的共享数据版本、其它协同设计者设计版本.再将所找到的各版本分别对其进行版本信息恢复操作.

如上例中,若要把 B5 所对应的项目子设计版本作为当前版本设置,则需分别恢复 A3, B5, C5, D2 这样的几个有关版本的信息.版本恢复时还要进行一些操作:

(1) 检查是不是源版本,如若是源版本则可跳过恢复操作;

(2) 若不是源版本,则需要找到此版本前驱路径上最近的一个源版本,从这个源版本开始依据版本发展路径逐个版本进行恢复.

5.2 版本的合并

版本合并包括:设计者私有设计版本的合并,如图 1 中 B4 即是由 B2 和 B3 合并而产生;其二是设计者项目子设计版本的合并;再就是整个项目设计版本的合并.私有设计版本

的合并,在工程数据库系统 EDBMS 中已经被解决.下面主要考虑项目子设计版本的合并.此时涉及到设计者的私有设计版本和其他设计者的设计版本以及共享信息版本.在合并处理时要依据系统版本字典和各设计者的版本字典及版本差集索引来进行合并处理.其主要过程是:

(1)恢复被合并版本的信息.根据被合并版本自己的版本字典找到系统版本号,依此版本号再查到其他有关的版本.把各版本的数据恢复出来,并将其合并到一起(具体操作是依 dbk 进行,因为 dbk 是唯一的);

(2)对项目子设计版本进行合并.合并时要对被合并版本中的数据表(分同名表和不同名表)进行不同的处理.相同的要进行选择,并记录下选择的结果,以备将来恢复时使用;

(3)要对被合并版本中的 V,L 等特殊属性进行相关的处理等操作.

5.3 版本删除

版本删除主要是对设计者私有设计版本进行删除操作,在这些版本中存放有设计数据,而上层版本都是在此基础上通过有关结构实现的.基本的版本删除包括 2 种具体的删除操作:一是仅删除指定的一个私有设计版本,它会使版本的变化发展更清楚;另一种是不仅要删除指定的版本还要删除所有能删的后继版本.

(1)仅删除指定版本,此时要将该版本中变化了的信息纳入其所有后继版本中.具体的操作是:

检查用户权限,只有版本主人才可进行版本删除操作.将被删版本的有关数据分别插入到后继版本(对 L,V 等特殊属性要进行特别处理).删除该版本的数据.修改设计者版本字典,删除该版本的记录项.修改设计者版本链,保持版本的一致性.

```
algorithm: delete-version(version-name)
  (if (version-name is not valid) error
   if (version-name is a key version)
     {printf("The key version should not be deleted)
      error
     }
  check user rights
  if (no right)
    {printf("you do not use this version-name!")
     exit
    }
  search deleted version in version dictionary
  if (found)
    {lock system data and this version
     find out all versions that will be deleted
     recover all data of relationship version
     related data insert to successive versions
     adjust version graphics
     modify version dictionary
     unlock system data
    }
```

}

(2)删除指定版本及所有能删的后继版本

检查用户权限,只有版本主人才可进行删除操作.从该版本出发遍历版本图,列出完全源于该版本的后继版本和不完全源于该版本的后继版本.对于不完全源于该版本的后继版本,则需要将被删除的父版本的有关数据插入其中.删除完全源于该版本的后继版本.修改设计者的版本字典,删除该版本的记录项.修改设计者版本链,保持版本的一致性.

6 结 论

计算机支持下的协同设计是一个很好的 CAD 环境.为了有效的支持协同设计,使设计者能较好地共享设计信息以及协调地进行设计,协同设计的版本管理是一个重要而困难的问题.本文就支持协同设计的多库分布式工程数据库系统的版本管理进行了研究,并进行了实现.在我们的系统中采用关系与网状混合的模型,并吸取面向对象的思想,从而兼备了传统数据库系统和面向对象数据库系统的优点,它的设计思想更符合于工程应用的特点.多级版本管理保证了设计信息的一致性,并且层次清楚,易于接受,便于应用.本系统正用于某 CSCW 项目的底层支持,并将为某商业 CAD 集成系统提供支持.对尚存在的一些问题也正在继续完善和提高中.

参考文献

- 1 Kyng M. Designing for cooperation: cooperating in design. *Communications of the ACM*, 1991, 34(12):64~73.
- 2 郭玉钊,林宗楷.关系/网状混合模型的 CAD 数据库系统. *计算机学报*, 1988, 7(7):395~400.
- 3 郑卫东,林宗楷,郭玉钊.工程数据的版本管理. *计算机辅助设计与图形学学报*, 1993, 5(4):292~296.

VERSION CONTROL OF EDBMS FOR COMPUTER SUPPORTED COOPERATIVE DESIGN

Wang Qianping Lin Zongkai Guo Yuchai

(CAD Laboratory Institute of Computing Technology The Chinese Academy of Sciences Beijing 100080)

Abstract Computer supported cooperative design is an important field in CAD. This paper discusses version control of EDBMS for cooperative design. Its feature and implementation method are also analyzed.

Key words Version control, CSCW, EDBMS, CAD, distributed.