

工程数据库中支持协同设计的版本管理

吴会松 (华北石油勘察设计院计算机室 任邱 062552)

摘要:在当前的工程数据库应用中,往往涉及了多个工程技术人员的配合问题,这就要求工程数据库必须支持协同设计。本文在探讨计算机支持协同设计的基础上,提出了一种用于支持协同设计的多库分布式工程数据库版本管理的基本策略和模型。

关键词:计算机软件 工程数据库 版本管理 协同设计 分布式

1. 引言

在实际的工程设计中,不仅需要很多工程师进行分工设计、协同工作,而且还要经常参考管理人员、施工人员和用户的意见,才能获得满意的设计效果。因此,使工程数据库系统具有支持协同设计的能力,已成为人们关注的一个焦点。鉴于此,本文对计算机支持的协同设计进行了探讨,并提出了一种用于支持协同设计的多库分布式工程数据库版本管理的基本策略和模型。

2. 计算机支持的协同设计

这是指能支持相关人员共同进行设计工作的一种计算机环境,它以群体工作目标为核心,让各方人员在充分发挥作用的前提下协同工作,对于保证设计工作的质量及速度极为有利。

参加协同设计的人员不仅有设计人员,还有不同部门的管理人员和最终用户。这些人员可能分布在不同的

地理位置,使用不同的计算机系统,但他们可以通过计算机网络相互协调,有效地保证设计质量和效率。

传统的设计方法往往会造成项目管理与设计之间、不同设计阶段之间、设计人员与施工者之间的脱节,这种方式会产生设计费用高、设计周期长、设计质量难以保证等缺点。而计算机支持下的协同设计环境将会大大改善这种局面。

协同设计要求数据管理必须有效,因此,具有能支持协同设计的、并能有效处理工程数据的分布式工程数据库管理系统是必不可少的支持软件,它也是建立集成化协同式CAD/CAM系统的核心。通过该系统可以实现各工作阶段的信息共享、协调各方的工作,使各方人员能够协同工作,最终使各项指标得以提高。

3. 分布式工程数据库管理系统

工程数据 © 中国科学院软件研究所 <http://www.c-s-a.org.cn>

统,目前在机械、建筑、集成化电子及油气田工厂设计等CAD/CAM 范畴内使用。因为缺少对协同设计的支持,其应用效果并不理想。

近来投入实用的分布式工程数据库管理系统对协同设计的支持情况大有改善。它除了具有传统工程数据库管理系统所具有的功能外,还具有一些特殊功能,如:支持工程长事务处理;支持对复杂对象之间关系(记录之间、表与记录之间、表之间)的描述;支持基于客户/服务器环境的分布式管理;支持数据模式的动态修改。另外在数据属性上也有相当大的扩充,如支持长记录的 L 属性,它可以支持工程中长记录的表示和处理;支持变长的 V 属性,它可支持变长记录的表示和处理。通过这些属性就可以支持对一些非传统数据类型的管理(如现在已很常见的多媒体数据)。这种新式数据库管理系统为支持协同设计提供了有益的基础。

4. 协同设计对版本管理的要求

协同设计要求一个工程(项目)的多个设计人员及管理者和用户能在网络环境下协同工作,这就对与之相关的工程数据库的版本管理提出了相当复杂的要求。

(1)必须能保留设计的历史。在实际的工程设计中,因为各个设计人员的知识、经验、背景及看问题的角度不同,对同一个设计对象往往会有不同的设计方案,因而会产生不同的设计版本。而一个好的设计往往是在对多个设计方案(即设计版本)进行分析、比较、综合、优化后才能完成。因此,协同设计就需要在工作中不断综合各设计人员的工作情况,不断地进行完善、优化。这就要求系统不仅能保留最终的设计结果,还要保留设计过程及设计中有关的方法和信息,即要求版本管理能够保留并管理多种设计方案及不断变化的设计模式和设计历史。

(2)能够管理分布于整个网络的大信息量。在网络环境下,与设计有关的信息和方法分布于服务器和客户机上。一般情况下,公用内容多放在服务器上,让大家共享。而设计人员的私有信息则放在自己的客户机上,供设计者自己使用。为了协同工作,有时需要将部分私有设计信息向其他设计人员开放,这样就使设计信息分散在整个网络上。在协同中,设计者之间往往会进行多媒体的交流(其中声、图的量最大),这些都会导致信息量大增。这就要求版本管理部分能够管理分布于整个网络的巨量信息,并使信息冗余小、存取速度快、管理效率高。

5. 支持协同设计的版本管理的基本策略

根据网络环境下协同设计中版本管理的特点,吸取面

向对象的基本思想,再参考整体-一部分的结构原则,可以采用多库多级版本管理的策略。

(1)用多库存储的方法化解技术难度。在复杂的系统中把信息分成多个数据库来进行管理是一种行之有效的办法,也是使数据结构规范化的基本策略。数据结构不同,分库的方法也不同。一般可以把公用的基本内容做成系统数据库放在服务器上让大家共享,把设计者的设计信息放在客户机上做其私有库,仅让其他人员有条件地共享其私有内容。这种划分数据库的方法可以借用现在已很成熟的单库管理技术,可以减少版本信息的冗余、减少总体信息的变化,进而减少版本的演绎。因为各库的内容较少,因而修改、合并、删除等版本操作需处理的信息量也少,处理也容易些。

(2)用分级的办法进行版本管理。因为在网络环境下进行协同设计时所管理的内容相当多,为了提高管理效率,我们可以采用分层的办法进行版本管理。一般可以分成以下几个层次:

①对设计者独自进行封闭设计时的版本管理是最基本的管理,这类管理有两个,一是对设计人员在其客户机上所作(封闭)设计的版本管理,二是对系统共享信息的版本管理。

②对设计人员之间有交流时的版本管理,即对设计者调用系统信息及其他设计者的私有信息进行设计时的版本管理。

③对整个项目的设计做版本管理。

显然,这种分级管理的方法可以降低管理的难度,并能在应用中逐步完善。

6. 支持协同设计的版本管理的基本模型

将设计信息分成多个库后,参照面向对象的思想,我们可以把每个库视为一个广义的类,这样就能把各个库的信息进行封装,比如把设计人员的私有设计信息封装在设计者的私有版本内,把系统信息封装在系统信息版本内。这种结合相关内容分类的方法使各部分的交叉大大减少,有助于保证设计信息的一致性,使变化程度不同的设计版本相互分离,因而可以进行有重点的版本管理。显然,重点应该在对设计者的版本管理上。

因为人的创造性思维多是在过去已有内容基础上的再提高,是个渐进的过程,所以在一般情况下,一个项目的(子)设计是私有设计与其他设计版本相结合的产物(真正无重复的完全创新设计极为少见),因而可以采用基于

多继承的多源有向无循环图的管理模型,以凸显其共有属性,提高信息的共享程度。具体到(子)设计的版本管理上,就是在基本的版本管理的基础上,扩充一些数据结构,再以此为基础进行版本管理。为了保证一致性,还应在分类的基础上配置数据字典(也属于门结构的范畴),如系统版本字典、设计者版本字典等。系统版本字典保存了当前系统版本的变化情况,属共享资源,应放在服务器上。系统版本字典是一切版本的基础。设计者若想产生新的设计版本,就得先向系统版本字典申请,在系统版本字典的统一控制下产生新的版本。

对于共享信息版本字典的修改,必须经授权后才能进行。设计者版本字典的修改最好由其本人自己进行,系统版本字典的操作,应在设计者后由系统进行。这种方法比较保险,而且简洁,能够满足绝大多数工程的需要。其缺点是对系统的并发性有一定程度的限制。

考虑到对多媒体信息的处理,可采用目前已很普及的通用型字段或 V、L 属性的方法,同时使用压缩存储。

在实际的工程设计中,前、后版本间的变化并不大,我们可以利用这一点,采用存储版本差集的方法减少信息的存储量:从一个具有完整信息的源版本开始,在版本的变化过程中,只保存与上一个版本间的差(即后一个版本只保存更新了的数据),并以记录为基本的存储单位。这种方法一般可节省 70% 以上的存储空间(不包括压缩程序的作用)。

7. 版本管理的基本功能及其基本的实现过程

(1) 选定当前版本。选定当前版本,就是把某人的某个项目的子设计版本定为当前要处理的版本。其基本过程是:先根据选定的版本检查用户的权限(一般只允许该版本的主人对它做这种操作);然后根据设计者版本字典中的版本号确定系统版本号,再根据系统版本号找到所包含的各种版本(如共享版本、其他协同人的设计版本等),并根据版本差集恢复这些版本的信息。

对于图 1 所示的例子,若要把 A2 所对应的项目子设计版本定为当前版本,就需要分别恢复 A2、B3、C5、G3 这几个版本的信息。恢复版本一般需要做以下的操作:

检查是否源版本,若不是源版本,就应从此版本的前驱路径上找到最近一个源版本,从这个版本开始向后逐个地恢复需要恢复的版本;若是源版本,则不作恢复处理。

(2) 合并版本。这里允许各种类型版本的合并。合并

时所依据的内容包括有关设计者的版本字典、系统版本字典及版本差集,其主要过程为:

① 恢复待合并版本的信息:根据待合并版本的版本字典确定系统版本号,据此找到其他有关的版本,根据版本差值恢复这些版本的信息。最后根据“数据库主码”将其合并到一起。

② 对待合并版本中的数据表和 V、L 等内容进行特殊处理。

(3) 删除版本。因为版本间可能存在着比较复杂的关系,所以删除时要考虑两种不同的类型:一种是只删除一个指定的私有设计版本;另一种是除删去一个指定的私有设计版本外,还要删除所有能删除的后继版本。

① 只删除一个指定版本。为了保证后继版本的信息完整,这时要把变化了的信息放入所有与之相关的后继版本中。基本过程为:先根据选定的版本检查用户的权限(一般只允许该版本的主人对它做这种操作);然后把该版本的有关数据分别插入与之相关的后继版本;删除该版本的数据;从设计者版本字典中删除该版本的记录;修改相关的版本链(保持系统的一致性)。

② 删除指定版本及与之相关的后继版本。先确定操作者的权限(一般只允许版本主人作此删除操作)。以该版本为起点向后遍历各相关版本,找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。对于前者,要完全删除;对于后者,则把待删除的父版本的有关数据放入版本内。做完这些后,再修改该设计者的版本字典,从中删除该版本的记录。最后修改相关的版本链(保持系统版本的一致性)。

参考文献

- [1] 郭钰钗,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):395400.
- [2] Kyng M. Designing for cooperation: cooperation in design. communications of the ACM, 1991, 34(12): 64-73.
- [3] 顾宁等.多表多记录存储管理及其应用.计算机辅助设计与图形学学报,1997,9(1):2934.
- [4] 陈其明.工程数据库原理.北京测绘出版社,1991.

(来稿时间:1997年4月)