

协同图形编辑系统多版本动态属性锁协议研究

徐牡莲, 李陶深

(广西大学 计算机与电子信息学院, 广西 南宁 530004)

摘要:在协同 CAD 系统中, 解决设计者之间的冲突检测与消解问题是实现有效并发控制机制的关键。针对协同设计中多个关联对象之间的冲突检测和消解问题, 介绍协同多版本相关概念和多版本属性标记锁策略, 提出一种解决对象关联操作冲突的多版本动态属性锁并发控制协议, 设计实现相应的算法。该协议将多版本和对象标记锁相结合, 保证了并发冲突中的用户意愿。最后, 分析多版本动态属性共享锁协议的正确性, 并通过实例说明协议的可用性。

关键词:协同图形编辑系统; 多版本; 动态属性; 共享锁; 并发控制; 冲突检测; 冲突消解

中图分类号: TP391.72

文献标识码: A

文章编号: 1673-629X(2008)12-0102-04

Research on Protocol with Multi-version Dynamic Attribute Lock in Collaborative Graphics Editing Systems

XU Mu-lian, LI Tao-shen

(School of Computer, Electronics and Information, Guangxi University, Nanning 530004, China)

Abstract: In collaborative CAD system, the key to the implementation of the concurrent control mechanism lies in solving the problem of conflict detection and resolution between designers. In view of existing problem of conflict detection and resolution between multiple object-related, the conception of the collaborative multi-version and the strategy of shared marked lock with multi-versioning dynamic attribute are presented in the paper. Then, a concurrent control protocol with multi-version dynamic attribute lock to solve object-relative operation conflict and corresponding algorithm are proposed. The protocol combines multi-version and object marked lock, and volunteers the user's intention-preservation of concurrent conflict. Finally, the correctness of shared lock protocol is analyzed and the availability the protocol is represented by instances.

Key words: collaborative graphics editing systems; multi-version; dynamic attribute; shared lock; concurrent control; conflict detection; conflict resolution

0 引言

现有的二维协同图形编辑系统主要有集中式和复制式两种体系结构^[1]。集中式系统统一由集中服务器管理并发操作和用户信息, 结构简单、并发控制容易, 但网络负载重、通信延迟时间长, 中心服务器容易造成系统瘫痪。复制式系统具有网络负载轻、交互功能强、响应速度快等特点, 但其一致性保证和并发控制变得相当复杂。因此, 研究全复制结构下的实时协同编辑系统中多副本一致性保证成为其关键核心问题。

Sun Chengzheng 等人首次提出一致性保证包括收

敛、因果维护和意愿保证, 提出一个全新的基于对象复制的广义多版本模型和处理策略^[2]。Liyin Xue 等人提出新的多版本合并策略, 根据用户的语义保证用户操作意愿^[3]。Sandy Citro 等人提出将操作转换和多版本结合一起解决不同类型操作冲突, 通过加用户意图锁解决部分意图冲突, 控制版本无限增长^[4]。但这些方法所解决的冲突都是针对操作对象相互独立的情况, 每次只作用于一个对象。事实上, 在各种机械制图中, 很多图形对象之间存在相互关联, 即本地用户在修改一个对象时, 如果协同用户正在修改另一对象, 且这两对象之间存在某种约束关联关系, 则它们之间可能会导致冲突。

目前解决操作冲突的方法大致分四类: 锁机制、序列化、dOPT 和对对象复制等^[5]。锁机制要求在编辑对象之前给对象加锁, 但需要一个集中服务器管理锁和协调锁的申请和释放, 从而导致网络瓶颈。序列化机制允许本地操作立即执行, 而远程操作需与该站点的

收稿日期: 2008-03-21

基金项目: 广西科技创新能力与条件建设项目(桂科能 07109008-006Z)

作者简介: 徐牡莲(1981-), 女, 湖南人, 硕士研究生, 研究方向为协同 CAD 系统; 李陶深, 教授, CCF 高级会员, 研究方向为 CAD 理论与应用、计算机网络、分布式数据库。

历史队列中已执行操作比较,确定是否存在冲突,若存在冲突,则系统将重新排序以确定哪个操作效果最终在视图中得到表现^[6]。分布式操作转换(dOPT)允许所有用户的协同操作在本地立即执行,同时将操作传递给其它节点,各节点接受到操作后,需与系统状态进行比较,必要时将操作做一定转换后再执行该操作^[2]。对象复制将原有对象拷贝多个副本,用于冲突操作,因而保证操作意愿,但对复杂操作难以实现。

文中研究关联对象之间的操作冲突控制和消解问题,提出一套将多版本和对象标记锁结合起来解决对象关联操作冲突的并发控制策略。

1 协同多版本的相关概念

多版本技术强调操作意愿的保证,当发生冲突操作时,采用版本复制和对象复制两种方式保证操作意愿。对象复制技术可以很好地应用于对象独立之间的并发冲突操作。而版本复制可以用于关联对象的并发冲突操作。当操作发生冲突时,从目标文档派生出两个子版本,分别应用冲突操作到不同版本中,这种方法叫做版本复制。

定义 1:操作 O (Operation)。将操作 O 定义为八元组 $\{ID, ROG, CT, Pr, AT, AV, AVT, R\}$ 。其中 ID 为操作对象的唯一标识, ROG 为操作关联对象组, CT 为操作类型, Pr 为操作优先权, AT 为对象属性的类型状态, AV 为对象属性值, AVT 为该操作对象属性锁标记, R 为发起操作的用户角色。

定义 2:因果关系。假设在节点 i 和 j 上生成两个操作 O_1 和 O_2 , 如果符合下面条件之一, 则说明 O_1 和 O_2 存在因果依赖关系: 记为 $O_1 \rightarrow O_2$ 。

- (a) $i = j$ 且操作 O_1 是在操作 O_2 之前生成;
- (b) $i \neq j$, 且 O_1 在节点 j 执行过程发生在操作 O_2 生成之前;
- (c) 存在一个操作 O_x , 且有 $O_1 \rightarrow O_x$ 和 $O_x \rightarrow O_2$ 。

定义 3:并发关系。设两个任意操作 O_1 和 O_2 不存在 $O_1 \rightarrow O_2$, 也不存在 $O_2 \rightarrow O_1$, 则 O_1 和 O_2 存在并发关系, 即 $O_1 \parallel O_2$ 。

显然,操作执行的因果关系或并发关系都与操作执行的时间顺序有关。

定义 4:操作意愿保留(Intention-preservation)。操作 O 的操作意愿是指 O 的执行效果与在产生 O 的文档状态下应用 O 的效果一致。对于任何操作 O , 在所有节点执行 O 的结果都应与 O 的意愿一致, 而且 O 的应用不能改变其它并发操作的执行效果。

协同编辑系统支持多个用户协作的进行图形对象的创建和修改等操作。操作意愿保留要求操作在远程

节点执行的效果与本地生成时执行的效果完全一致。因为操作意愿与系统上下文有着密切的关系, 所以操作意愿保留的实现非常复杂。

协同多版本技术下, 所有节点的共享文档由一组物理版本来表示。其中每个版本表示为一组具有不同属性的图形对象, 且各对象是相互关联的。在实时协同环境下, 多个节点并发操作必然导致冲突操作。因此, 当某一用户对对象关联集中的对象进行修改时, 则关联对象的同一属性也跟随修改。

定义 5:对象关联度和对象关联组。对象关联度是指对象与其它对象的具有关联关系的对象的数目, 记为 RD 。关联对象集是指对象与其具有关联关系的对象的组合, 记为 ROG 。

2 多版本属性标记锁策略

2.1 相关概念定义及表示

当冲突操作产生来自于修改某对象组中同一属性时, 则当前关联对象组产生冲突。可采用多版本属性锁协议, 最后由用户操作管理者对最终版本确定, 即第三方仲裁机制确定最后版本。

定义 6:锁集。当某操作 O 针对对象 G 操作时, 则根据操作 O 的 AT 找出关联对象组, 对该关联对象组中所有对象加锁, 被加锁的对象集合称为锁集。记为 $LG(O)$ 。

定义 7:几何冲突操作与相容操作。假设操作对象之间有关联, 两操作 O_1 和 O_2 分别对具有关联对象进行操作, 且满足:

- (1) $O_1 \parallel O_2$
- (2) $O_1 \cdot ID, O_2 \cdot ID \in ROG_i$ (关联对象组)
- (3) $O_1 \cdot AT, O_2 \cdot AT \in GS$ (几何属性)
- (4) $O_1 \cdot AT = O_2 \cdot AT$
- (5) $O_1 \cdot AV \neq O_2 \cdot AV$

则称 O_1 和 O_2 之间存在几何冲突, 记为 $O_1 \oplus O_2$;

若 O_1 与 O_2 之间不存在几何冲突关系, 则称它们之间几何相容, 记为 $O_1 \odot O_2$ 。

2.2 关联对象组集动态运算规则与算法

关联对象组集即由所有关联对象组构成的集合, 当用户操作为插入对象或删除对象时才引起关联对象组集的更新。

产生关联对象组集要符合下面三条规则:

规则 1:关联对象组之间相互独立。

规则 2:任何对象只能属于一个关联对象组。

规则 3:关联对象组集包含操作中的所有对象。

定理 1:通过上述三条规则产生的关联对象组集

(ROGS)是唯一的。

证明:用反证法。假设存在两个不同关联对象组集 $ROGS_1$ 和 $ROGS_2$ ($ROGS_1 \neq ROGS_2$)。则两个关联对象组集之间至少存在两组不相同的 $ROG_i \neq ROG_j$, 且 $ROG_i \subseteq ROGS_1, ROG_j \subseteq ROGS_2$; 假设不相同的元素为对象 $G \in ROG_i$, 但 $G \notin ROG_j$; 根据规则 3 得 G 必定属于 $ROGS_2$ 中 ROG_k ($k \neq j$) 对象, 且 $ROG_i \neq ROG_k$; 则 G 同时存在两个不同的关联对象组, 与规则 1 和规则 2 相违背。则说明原来的假设不成立, 即原命题成立。证毕。

关联对象组集生成算法如下:

Step1: 初始化 $ROGS = \{ROG_1, ROG_2, ROG_3, \dots, ROG_k\}$;

Step2: 当用户 j 产生第 i 个操作 O_i 为创建对象 G_m 操作, 创建对象标志为 $G_{mid} = \{U_{id}, i\}$ 时, 如果该对象与已存在某对象属性有关联关系或约束关系, 则将其加入到相应的关联对象组 ROG_t ($1 \leq t \leq k$) 中; 若该对象是独立对象, 即不与任何对象相关联, 则将该对象作为一个独立关联对象组 ROG_{k+1} , 并加入到关联对象组集 $ROGS$ 中。

Step3: 当用户 j 产生第 i 个操作 O_i 为删除对象操作, 则包含该对象的对应关联对象组也要删除该对象, 即更新 $ROGS$ 。

Step4: 当用户 j 产生第 i 个操作 O_i 为修改对象操作时, 则关联对象组中所有对象同时修改, 但并不影响关联对象个数, 因此, 关联对象组集 $ROGS$ 不变。

3 多版本动态属性共享锁协议

3.1 协议的描述

多版本属性共享锁协议的内容描述如下:

(1) 操作参与者(协同站点)。

若操作为本地修改操作 O_1 , 则将计算操作锁集 $LG(O_1)$ 并检查属性锁值, 做以下操作:

1) 若该属性锁值 ≥ 1 , 则在本地站点产生新的临时工作版本, 并在其上更新。更新后将最终结果和属性锁值发送给操作管理者, 不修改属性锁值。

2) 当操作的属性共享锁值为零, 表示该属性当前无人修改, 则对该对象锁集中所有对象全部加锁, 并将属性锁值加 1, 并产生一临时新版本。更新后, 将该版本及属性锁值发送到操作管理者, 不修改属性锁值。

(2) 操作管理者(协同小组组长)。

操作管理者接收到操作参与者的更新数据或临时版本时, 则临时版本计数器加 1, 比较版本计数器与该版本操作属性的值, 若相等, 表明对该属性操作的所有

版本接收完毕。然后再依情况处理:

1) 若临时版本计数器值为 1, 则表明只有一个用户修改该属性, 没有产生冲突版本, 将临时版本计数器值属性锁标记值清零; 将版本时间戳加 1, 将该版本设定为最终版本, 然后发送给各协同站点。

2) 若临时版本计数器值大于 1, 则表明产生了多个冲突操作版本。操作管理者采用第三方仲裁机制, 即根据一定算法策略选择并确定最终版本, 版本时间戳加 1, 属性锁标记值清零, 并将最终版本发放到所有参与协同用户。

3) 各用户接受更新版本, 并删除所有缓存中的临时版本。

在该协议中, 当操作产生冲突时, 都在自己复制版本中操作修改信息, 由协同管理者负责各站点一致性维护, 而各站点只维护当前临时版本信息, 因此不存在中间目标版本选择问题, 版本管理相对简单, 不会产生版本树增长。但是这需要牺牲协同管理站点维护共享界面的一致性, 并且对网络要求比较高, 一旦网络瘫痪, 冲突版本就得不到解决。

3.2 协议的正确性说明

从以下三个方面来说明该协议的正确性:

(1) 该协议能检测到冲突。协议将共享锁和临时版本标记相结合, 共享锁标记值代表了冲突操作用户数, 而操作管理者接收的对同一属性修改接收的临时版本数, 也代表用户做该属性修改的用户数。当数值等于一时, 代表只有一个用户操作该属性, 根据冲突定义, 不产生冲突。当数值大于 1 时, 则至少有两个用户在同时修改该属性值, 代表有冲突。因而能检测到冲突。

(2) 冲突解决的有效性。该协议是通过操作管理者对冲突多版本结果进行仲裁机制。当正确检测到冲突后, 各站点将产生临时版本, 并将临时版本发送给操作管理者实行仲裁, 仲裁的最终结果也将发送到各用户。因此, 操作管理者可以有效地解决各冲突操作和冲突版本。

(3) 保证共享界面能实现一致性维护。该协议由操作管理者负责所有站点的操作确认和更新操作版本, 并由操作管理者统一播放最终版本, 因此各站点的结果一致性能得到保证。由于冲突操作都分别在各自站点临时版本中更新操作, 每个站点都保留着本地操作意愿和本地操作执行后的效果, 因此能保证各协同用户的操作意愿。

3.3 协议的实例性验证

假设三个用户 U_1 、 U_2 和 U_3 针对一组关联对象组 $ROG = \{G_1, G_2, G_3\}$ 对象做修改操作: 用户 U_1 将

对象 G_1 右移 50cm, 用户 U_2 将对象 G_2 左移 50cm, 用户 U_3 将对象 G_3 上移 30cm(如图 1、2、3、4 所示)。用户 U_1 、 U_2 、 U_3 都是针对关联对象组中不同对象的位置属性做修改操作, 因为关联对象组之间有相对固定位置关系, 所以它们之间产生冲突。

假设共享对象组接受三个用户对位置关系的修改。当用户 U_1 先访问位置属性共享锁标记, 此时标记值为零, 故 U_1 为第一个修改用户, 它将其锁集中所有对象属性标记加 1, 并在本地站点产生复制临时版本 V_1 , 最后将版本 V_1 发送给操作管理者。

由于并发关系, 用户 U_2 和 U_3 同时也访问锁集标记锁值, 并将标记锁值累加 1 操作, 分别产生两个临时版本 V_2 和 V_3 (如图 3 和 4 所示), 并将 V_2 和 V_3 分别发送给协同管理者。

对于操作管理者来说, 接受版本 V_1 、 V_2 和 V_3 。每当接收到一个临时版本时, 操作管理者将临时版本计数器加 1, 当临时版本数与属性锁标记值都为 3 时, 表明所有临时版本接收完毕。最后由操作管理者实行仲裁, 决定采用哪个版本为最终版本, 并将最终版本发送到 U_1 、 U_2 和 U_3 的本地界面窗口中。

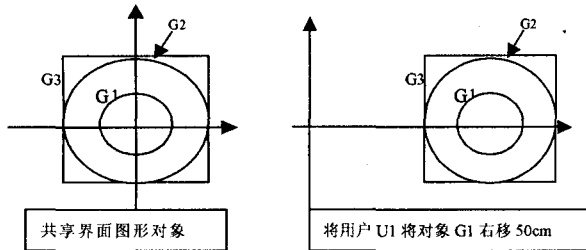
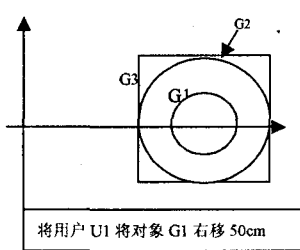
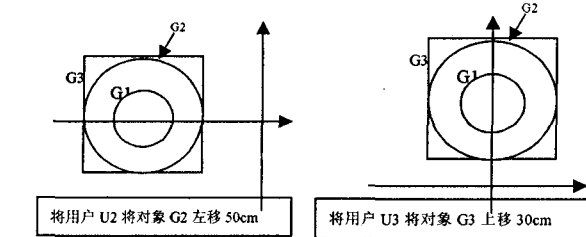
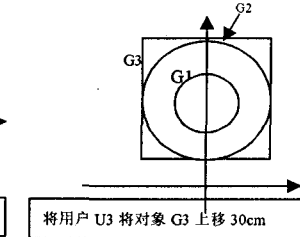


图 1 共享对象

图 2 U_1 用户操作后界面图 3 U_2 用户操作后界面图 4 U_3 用户操作后界面

(上接第 101 页)

参考文献:

- [1] Hecht - Nielsen R. Neurocomputer Applications[J]. Neural Computer, 1988, 2(1): 455 - 453.
- [2] 李士勇. 模糊控制·神经控制·智能控制论[M]. 哈尔滨: 哈尔滨工业大学出版社, 1996.
- [3] Piovoso M J, Kosanovich K A. Application of Multi - Variable Statistical Methods to Process Monitoring and Controller Design[J]. Int J. Control, 1994, 59(3): 734 - 765.

4 结束语

提出了一种解决对象关联操作冲突的多版本动态属性锁并发控制协议, 设计实现了相应的算法。该协议利用操作关联对象集, 检测当多个用户同时修改关联对象组同一个属性产生不同值时发生的操作冲突情况, 解决了一次操作针对在多个对象体现操作效果的操作冲突。分析了协议的正确性, 并通过实例说明该协议能很好地解决操作关联冲突。但该协议所有版本都通过服务器控制版本, 虽然可以使得版本和并发控制得到保证, 但是也造成了服务器与客户器之间频繁通信, 容易导致网络瓶颈。另外, 当版本过多时, 如何智能化地选择版本是多版本技术中有待研究的问题。

参考文献:

- [1] 高曙明, 何发智. 分布式协同设计技术综述[J]. 计算机辅助设计与图形学学报, 2004, 16(2): 149 - 157.
- [2] Sun C Z, Chen D. Consistency Maintenance in Real - time Collaborative Graphics Editing Systems[J]. ACM Transactions on Computer - Human Interaction, 2002, 9(1): 1 - 41.
- [3] Xue Liyin, Orgun M, Zhang Kang. A Multi - versioning Algorithm for Intention Preservation in Distributed Real - time Group Editors[C] // Twenty - Fifth Australasian Computer Science Conference(ACSC2003). Adelaide, Australia: [s. n.], 2003: 19 - 28.
- [4] Citro S, McGovern J, Ryan C. Conflict Management For Real - Time Collaborative Editing in Mobile Replicated Architectures[C] // Proceedings of the Thirtieth Australasian Conference on Computer Science. Ballarat, Australia: [s. n.], 2007.
- [5] Kimwoosacng, Srivastava Jaidee P. Enhancing real - time DBMS Performance with multiversion data and Priority based disk scheduling[C] // Proceedings of the 12th Real - time Systems Symposium. Neworlean: [s. n.], 1991: 222 - 231.
- [6] Sun C Z, Jia X, Zhang Y, et al. Achieving convergence, causality - preservation, and intention - preservation in real time cooperative editing systems[J]. ACM Transaction on Computer - Human Interactions, 1998, 5(1): 59 - 68.

- [4] 尚长军, 陈曦, 钱积新. 基于 PCA - BP 神经网络的精馏塔产品组成软测量模型[J]. 工业仪表与自动化装置, 2003(4): 33 - 36.
- [5] 刘克显. 软测量及智能控制在复杂工业过程参数检测与控制中的应用研究[D]. 沈阳: 东北大学, 2001.
- [6] 贾群, 李和平. 软测量与工控组态软件在 DMF 微量水工艺中的应用研究[J]. 中国仪器仪表, 2007(10): 47 - 50.
- [7] 贾群. OPC 技术在过程工业中的应用研究[J]. 中国仪器仪表, 2006(5): 72 - 74.