

# 基于日志转存的冲突消解策略

屈正庚, 卢琼, 赵玉霞

(商洛学院 计算机科学系, 陕西 商洛 726000)

**摘要:**协同编辑过程实质上就是并发操作的冲突产生与消解过程,有效合理地解决冲突是协同编辑过程的关键问题。首先分析了协同编辑过程中冲突产生的原因,然后对冲突分类汇总,利用时间和参数相关性防止冲突发生,提出了事务日志转存分组策略和事务关联树进行消解冲突,对协同模型重新组合与更新,而且在时间和空间上不受任何约束,最后通过一个事务建模过程进行应用。经过实验证明,采用日志转存方式保证了协同数据的一致性,很好地解决了协同冲突。

**关键词:**协同编辑;冲突消解;关联树;一致性

**中图分类号:**TP31

**文献标识码:**A

**文章编号:**1673-629X(2012)12-0131-04

## Conflict Resolution Strategy Based on Log Dump

QU Zheng-geng, LU Qiong, ZHAO Yu-xia

(Department of Computer Science, Shangluo University, Shangluo 726000, China)

**Abstract:** Collaborative editing process is essentially concurrent operations of the conflict and digestion process, effective and reasonable solution to the conflict is the key issue of collaborative editing process. First, the causes of conflict in collaborative design process is analyzed, then summary the conflict classification, use correlation of time and parameters to prevent conflict, and put forward some affairs log dump grouping strategy and affairs associated tree for digestion conflict, for collaborative model to combination and renewal, and in time and space without any constraint, finally is applied through a process of modeling business. The experiments show that the log dump is to ensure consistency of data together, a good solution to the collaborative conflict.

**Key words:** collaborative editing; conflict resolution; associated tree; consistency

## 0 引言

协同设计就是通过计算机网络技术,把许多工作人员集中起来共同围绕同一个目标开展工作,最后得到满足需求的结果。在整个设计过程中,所有工作人员并行性、交互性地参与工作,他们之间相辅相成,缺一不可,共同努力完成任务。但是每个工作人员考虑问题的标准、所学的专业知识、理解设计的思路不同,必然会引起冲突的发生。因此,协同设计就是冲突产生与消解的反复过程。

在计算机网络环境的支持下,不同地区的工作人员同时处理一件事务,协同编辑同一个文档、图形、图像、动画、视频等对象,齐心协力、高效快捷地进行设计与观看<sup>[1,2]</sup>。所以协同编辑系统(Cooperative Editing Systems, CES)支持多用户并发性与交互性,让任何用户随时随地可以浏览与编辑事务对象的任何部分,没

有任何约束与限制。一般情况下, CES 分为集中式和复制式两种,集中式结构简单,并发操作容易方便,但是增加了网络的负载,使网络通信出现延时,造成“瓶颈”现象。复制式要求每个网络服务器都要放置事务编辑模型和共享对象的状态变化,使每个用户处理对象及时同步,具有网络负载轻,交互性与响应性好的特点,但是关键问题在于复制式 CES 如何保持多个复制视图对象的一致性。所以一致性维护成为 CES 研究的焦点<sup>[3,4]</sup>。当不同网络服务器生成的并发操作对象改变同一个对象的相同属性到不同属性值时,并发冲突问题就引起了,如两个用户同时改变同一个对象上的颜色值,改变顺序不一致就会引起文档的不一致。

冲突消解是解决 CES 一致性数据维护的基础。根据协同设计自身的特点,从事务的并发性、实时性、原子性、交互性等方面做了深入的研究,保证所有工作人员设计信息一致性,同时杜绝设计过程中冲突的发生,由此文中提出了一种冲突消解机制——日志转存的策略,在实际应用中得到了很好的效果。

## 1 协同编辑冲突的分类与管理

在协同编辑过程中,冲突的发生可能由于设计人

收稿日期:2012-04-16;修回日期:2012-07-21

基金项目:陕西省自然科学基金项目(11JK1067,12JK0747);商洛学院科学与技术研究项目(10SKY018)

作者简介:屈正庚(1982-),男,陕西汉中,人,硕士,讲师,研究方向为协同设计与网络控制。

员自身的差异性,也有可能是设计任务的需求分析、详细设计、方案修订,也有可能是组织部门之间的协调关系等等。冲突在协同编辑过程中起到两个方面作用:第一,能够得到促进作用。这个冲突的发生是下一个优秀设计的开始,冲突产生与消解终而复始,最终得到一个最优的产品诞生了。第二,能够得到消极作用。冲突的产生一定会引起用户信息的不一致性,造成设计过程的不和谐,既浪费了时间又增加了成本。常见的冲突类型有:知识冲突、资源冲突、计划冲突和数据冲突。

冲突管理就是在协同设计过程中预防冲突发生,保证任务高效率完成,它会牵扯到三个方面重要技术:冲突避免技术、冲突检测技术、冲突解决技术,它们相互影响、相互制约、紧密联系在一起,在整个设计中起到导航作用,如图 1 所示。

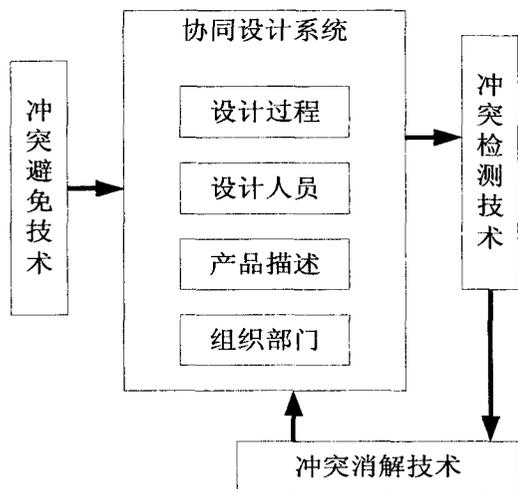


图 1 冲突管理

一般情况下,把冲突尽最大努力解决在初期。冲突避免技术就是冲突未发生之前采取一定措施防止一些冲突不发生。常见的冲突检测技术有基于真值、基于 Petri 网和不可满足约束的冲突检测方法。冲突解决技术就是针对无法避免的冲突采取一定技术可行性地化解。

## 2 冲突消解现状分析

引发冲突产生的原因很多,造成冲突类型也复杂,解决冲突任务艰巨困难。冲突消解在协同编辑过程中处于关键的地位,寻求最佳的方案来解决已经发生的冲突。国内外对冲突消解做了深入的研究与探讨,主要使用的技术有基于约束、基于规则、基于实例、基于协商策略。在设计过程中单个用户之间的冲突一般采取约束、规则和实例,集体冲突采取协商措施。

根据目前现状分析,常见的冲突解决策略主要从某一方面去解决问题,在实际设计中去应用却有局限

性、片面性。协同设计是一个动态变化的、复杂多样的、并行交互的过程,整个过程所发生的冲突根本无法用单一消解技术去化解矛盾,需要采用多种消解技术综合起来,科学合理地解决冲突。因此,文中采用了一种日志转存的方式消解事务冲突,设计了消解模型,详细描述了处理冲突的具体步骤。

## 3 日志转存的事务概述

一般情况下,事务处理过程采用 C/S 结构,网络服务器端存放共享对象的模型,用户端进行对象设计与编辑,并且把最终结果显示出来,网络起到信息传输与资源共享的作用<sup>[5]</sup>。但是网络的速度经常不能满足用户需求,所以采用复制式的体系结构进行操作对象,利用日志去处理事务,日志可以起到两个方面的作用:第一,可以减轻网络负载能力;第二,可以进行资源整合,快速地把操作对象传送给用户。

### 3.1 事务的日志描述

在协同编辑过程,多个用户同时参与工作,会产生多个事务,形成一个事务集合,对共享对象的事务集  $T(T_1 T_2 \dots T_i \dots T_j \in T)$  的日志进行描述:

- <  $T_i$ ,  $T\_Begin$ ,  $T\_Type$  >
- <  $T_i$ ,  $T\_StartID$ ,  $T\_EndID$  >
- <  $T_i$ ,  $T\_OperateID$  >
- <  $T_i$ ,  $T\_ParameterID$  >
- <  $T_i$ ,  $T\_CommitID$  >

在整个描述中,  $T_i$  称为事务集合  $T$  中随机一个事务,  $T\_Begin$  称为整个事务集开始时间,  $T\_Type$  称为整个事务集中所有事务的类型,  $T\_StartID$  称为当前事务开始时间,  $T\_EndID$  称为当前事务终止时间,  $T\_OperateID$  称为对当前事务进行所有操作的集合,  $T\_ParameterID$  称为当前事务的所有参数值设置情况,  $T\_CommitID$  表示对当前事务提交处理。

### 3.2 事务的操作过程

所有用户在协同编辑系统中都要对事务进行处理,不可避免地会引起冲突发生<sup>[6,7]</sup>。

为了更好地处理事务操作时发生的冲突,需要定义如下概念:

并发操作:参与设计者的所有用户同时对网络共享的对象进行操作,生成一个里面含有自己的操作属性集合  $D_i, D_i = (D\_startID, D\_endID, D\_actorID, D\_typeID)$ 。表达式中  $D\_startID$  为某一用户对共享对象进行操作起始时间,  $D\_endID$  为某一用户对共享对象进行操作完成时间,  $D\_actorID$  为协同操作某一用户,  $D\_typeID$  为某一用户改变共享对象操作的属性。

操作流:所有用户在协同编辑系统中对共享对象进行操作,经过不固定的时间段后会产生一个操作集

合,这个操作集合称为操作流。影响操作流主要有时间相关性、空间相关性、对象属性集这三个方面因素。

时间无关性:所有用户采用先来先服务的策略对共享对象进行操作,无论何时,任意用户对事务集中任何两个操作对象  $D_i$  和  $D_j$  进行操作,如果满足  $D\_startID(T_i) > D\_endID(T_j)$  或  $D\_endID(T_i) < D\_startID(T_j)$ ,此时这两个操作在时间上没有任何关系,必然对共享对象无法引起冲突<sup>[8]</sup>,否则就会产生冲突。

空间无关性:任意两个事务时间上有一定关联,如果满足  $S(D_i) \cap P(D_j) = NULL$ ,则称两个事务操作空间无关,一旦满足空间无关性,则这两个操作也不存在冲突。对于任何两个操作只有在时间上相关,同时满足  $S(D_i) \cap P(D_j) \neq NULL$ ,此时空间上才会发生冲突,引起操作分离,共享的事务呈现出不正确的状态。

参数无关性:在这里  $P(T)$  表示任意用户修改共享对象参数程度,所有用户在时间上和空间上存在关联,此时对共享对象进行操作,如果任何两个操作对象  $D_i$  和  $D_j$  在时间和空间上关联且满足  $P(T_i) \cap P(T_j) = NULL$ ,表示用户修改参数不相关,不会引起共享对象产生冲突;如果满足  $P(T_i) \cap P(T_j) \neq NULL$ ,此时两个事务立刻发生冲突。参数相关性引起冲突发生,会导致操作对象数据的错乱性,最终让所有用户信息的不正确<sup>[9]</sup>。

组合模型:整个协同编辑系统中,用户同时对操作对象进行操作,操作对象的状态时时刻刻在改变。假设经过  $\Delta t$  时间段后,操作对象呈现的状态是操作流状态,此时对所有的操作对象的状态进行排列组合与优化。如果用户越多,产生的操作流状态越多,必然引起排列组合与优化的事务数目会增多,这样不仅影响了协同编辑的并发性、交互性,而且大幅度地增加了网络的负载能力,降低了协同设计的工作效率,所以在实际协同编辑系统中,采用事务排列组合和相关性分组策略方式进行处理。

### 3.3 日志转存的事务处理模型

基于日志转存的事务处理模型,主要采用 C/S 模式进行日志转存,如图 2 所示。

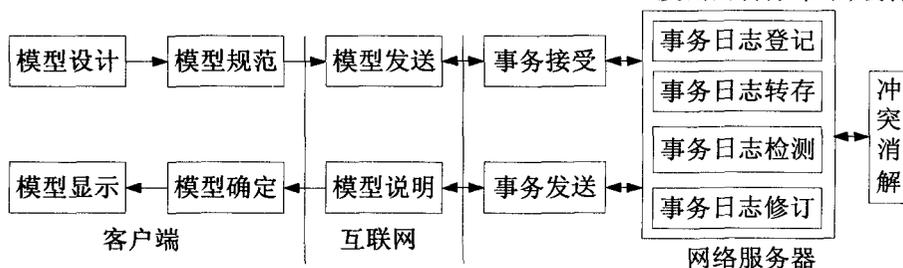


图 2 事务处理模型

客户端主要进行协同数据模型的设计与显示,让

所有设计者都可以看到自己的不足。互联网主要进行模型数据的传输与共享,保证信息的及时响应性和数据一致性。网络服务器主要对用户设计的模型进行登记、检测、修订,使得最终的数据和用户需求的数据一致,在这个过程中进行冲突检测与消解。

日志转存在这模型中起了三个方面作用:

第一,减轻了网络负载能力,让信息传输畅通无阻,参加设计者快速有效地得到消息。

第二,整合不同设计者传输过来的信息,组成一个完整有效的模型,及时对此模型进行解释说明,让每一个设计者明白自己的不足。

第三,如果协同编辑系统出现故障,能够帮助系统自我恢复。通过每次记录的日志找出最近一次没有故障的系统,保证了数据信息完整性。

### 4 日志转存的冲突检测与消解机制

整个协同设计系统中,所有用户参与工作不断改变操作对象的状态,必定产生很多事务,对这些事务进行排列组合肯定造成用户操作对象的不一致性,引起网络负载能力加强,此时采取事务分组策略来改变数据不一致性和网络延迟问题。

例如任意两个事务集合  $\{(t1, t2) \in T1\}$  和  $\{(t1, t2) \in T2\}$ , 这里如果  $MIN(D\_startID(T2)) > MAX(D\_endID(T1))$  或者  $MIN(D\_startID(T1)) > MAX(D\_endID(T2))$ ,称之为事务集合  $T1$  和  $T2$  在时间上没有任何关联。如果  $T1$  中任何一个事务与  $T2$  中任何一个事务既在时间上无关性,又在空间上无关性,则这两个事务集合随意改变顺序,不会产生任何冲突<sup>[10,11]</sup>。

在实际设计中,所有用户同时对共享对象进行操作,不可避免地会发生冲突,这时针对不同的状况采取不同的、有效的措施进行解决,具体执行的策略如下:

(1) 时间无关性。针对时间无关性采取用户强制分组策略或者协同编辑系统设制固定时间自动进行分组,让所有用户对共享对象操作在时间上无任何关系。参与操作的用户经过一段时间  $\Delta T$  后,听从队长发出的暂停命令,或者协同系统自动进入暂停状态,这

时所有用户无法进行任何操作,暂停这段时间就是对自己当前的状态进行补充与完善,保持数据一致性。

(2) 空间无关性。只要任何两个事务集合中的任意事务在时间上无关性,此时采取智能控制策略,通过网络传送信息时记录每一次的日志,通过清晰的日

志保证所有用户在操作不会产生冲突。

(3) 参数无关性。根据相关参数的相似性对所有事务进行分组与排列组合,构成一种事务关联图,具体情况如图 3 所示。所有用户对事务进行操作,产生各种各样的状态信息,依据参数相似度跟踪上一次构成的关联图,再次构成一个新的关联图出来。

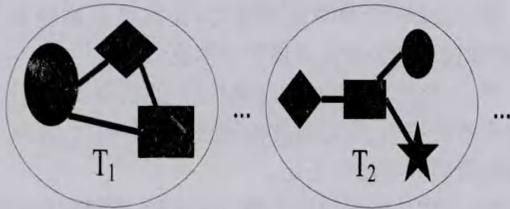


图 3 事务关联图

(4) 组合的事务模型。所有用户在网络上对共享对象进行操作都会产生一个操作记录,把这些操作记录传送给网络服务器,在服务器端形成一个有序可靠的队列。如果服务器端已经存在  $i - 1$  个事务日志,则新来的事务  $T_i$  产生的日志利用遍历树的方式归类到已经排列有序的队列中去。具体的操作过程如下:

第一步,对前  $i - 1$  个事务采用先来先服务的策略进行遍历,任意一个事务都要与新来的事务  $T_i$  进行关联性判定,如果存在关联,则产生一条线路,把所有事务遍历完就可以形成一种事务关联树,如图 4 所示。

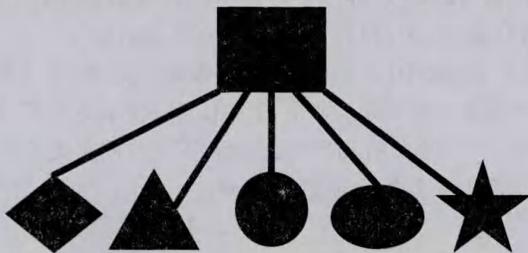


图 4 事务关联树

第二步,形成的关联树进行排列与组合,得到不同的事务序列。例如事务  $T_i$  与事务队列  $i - 1$  中有  $n$  个事务是关联的,这时把  $T_i$  归类到这  $n$  个事务集合中去,构建一个新的  $n + 1$  事务序列。

第三步,通过网络服务器把刚构建的  $n + 1$  事务新序列传输给用户端,此时用户端就会呈现出复杂多样的结果,因此,所有用户参与进来,利用网络进行协商与处理,选定出一个有效的、可靠的方案处理事务冲突,同时记录下目前的状态信息以及  $T_i$  事务所处的位置,保证数据完整性。

### 5 实际应用

在协同设计过程中,一般是从最基本的模型入手,所有设计者完成自己任务以后,对基本模型实现共享,在网络服务器中对基本模型进行组合,然后每个用户

分配任务进行编辑,最终得到确定的模型<sup>[12,13]</sup>。基于事务日志转存的冲突消解机制把每个用户操作的文档进行保留与整合,避免冲突发生,得到一个合理有效的结果。基本过程如图 5 所示。

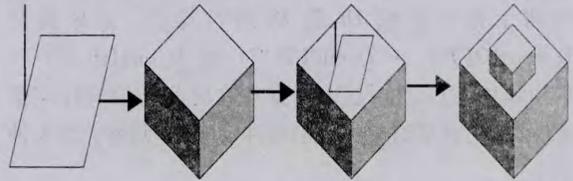


图 5 建模过程

实践证明,多个用户在协同编辑过程中采用事务日志转存的方式去处理冲突,得到了很好的效果,并且保证了数据的一致性,维护了事务的并发性与完整性。在各个网络服务器上得到相同的结果。

### 6 结束语

协同编辑过程中冲突消解占有重要的位置,很多学者对协同编辑做了大量的研究,但是在冲突消解方面仍然存在一些问题<sup>[14]</sup>。文中通过实际应用构建一个模型,在编辑系统中采用日志转存进行冲突消解,通过时间无关性和参数无关性的方法实现冲突检测,通过建立事务分组策略和事务关联树进行冲突消解,降低了网络负载与延迟,提高了工作效率,并且在实际应用中具有一定的价值与意义。

#### 参考文献:

- [1] 戚伟业,刘弘. 协同编辑系统中保持共享对象一致性的研究[J]. 计算机技术与发展,2008,18(3):63-66.
- [2] 孔令富,李晓梅,冯建周. 同步协同设计中并发控制的研究[J]. 计算机研究与发展,2009,46(2):346-350.
- [3] 徐牡莲,李陶深. 协同图形编辑系统多版本动态属性锁协议研究[J]. 计算机技术与发展,2008,18(12):102-105.
- [4] 李春萍,马瑜. 协同图形编辑系统的操作冲突与消解[J]. 玉林师范学院学报,2009,25(8):45-49.
- [5] 汪大勇,金炜东. 协同设计中的并行冲突检测算法[J]. 计算机应用,2007,27(3):650-652.
- [6] 徐牡莲,李陶深. 协同图形编辑系统中并发控制冲突检测的研究[J]. 计算机工程与科学,2009,31(6):44-46.
- [7] 路川,胡欣杰,纪锋. 基于角色访问控制的协同办公系统设计与实现[J]. 计算机技术与发展,2010,20(3):230-233.
- [8] 孙博,张荣国,王艳艳. 协同设计中的模糊冲突消解方法[J]. 计算机工程与应用,2009,45(3):84-86.
- [9] 吴月萍,王娜,马良. 基于蚁群算法的协同过滤推荐系统的研究[J]. 计算机技术与发展,2011,21(10):73-76.
- [10] 孟秀丽. 协同设计支持环境及冲突消解理论与方法[M]. 南京:东南大学出版社,2010.

图 4 为网络中被管设备中 ipInReceives(3) 对象的相应时间比较,从图上可以看出:算法刚启动的短时间内优先级排队算法并无明显优势,但经过一段时间后,优先级排队算法的时间响应明显小于一般排队算法的时间响应。

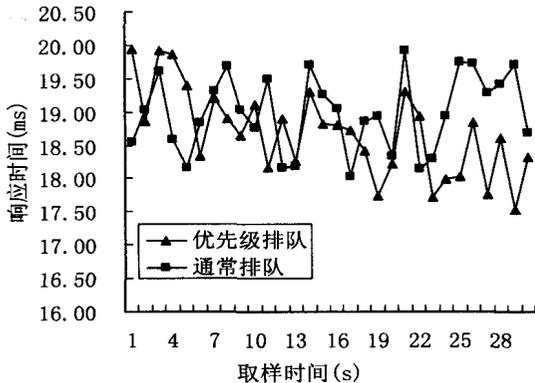


图 4 响应时间比较

综合图 3 和图 4,可以看出,文中所提出的算法,在减小网络管理过程中数据量和缩短访问对象的时间都有一定的作用。

### 3 结束语

作为网络管理的应用协议 SNMP 对于计算机网络管理发挥了重要作用,但对于动态的网络状态,SNMP 协议在很多方面就显得过于简单。在文中,提出了一种基于动态特征的统计方法,可以根据动态统计的结果修正轮询时间以及优先级从而使轮询过程不断优化。算法在初始状态下是根据相关配置进行周期以及优先级级别的设置,随着算法的运行,经过一段时间的自适应调整,这种轮询服务达到最优,另外,可以结合拥塞控制等相关软件接口,达到对网络性能动态调控的目的。

该算法也有不足之处,主要表现在:由于服务器要

实现轮询循环列表的修正并计算优先等级以及轮询周期,这些都运行在监控服务器上,可能对服务器造成一定的负担,这也是需要进一步研究的问题。

### 参考文献:

- [1] 孙 霞. 基于 SNMP 的远程服务管理模型研究[J]. 计算机时代,2009(4):33-34.
- [2] 史爱军. 基于 web 方式的 SNMP 网络管理实现[J]. 信息通信,2009(6):54-55.
- [3] Rose M T. Management Information Base for Version 2 of the Simple Network Management Protocol (SNMPv2) [S]. RFC1907,1992.
- [4] Harrington D, Presuhn R, Wijnen B. An Architecture for Describing SNMP Management Frameworks [S]. RFC2271, 1998.
- [5] 武孟军,徐 龚,任相臣. Visual C++开发基于 SNMP 的网络管理软件[M]. 北京:人民邮电出版社,2007.
- [6] 李明江. SNMP 简单网络管理协议[M]. 北京:电子工业出版社,2007.
- [7] Martin-Flatin P, Bovert L, Hubaux P. JAMAP: a web-based management platform for IP networks[C]//Proc. of the 10th IFIP/IEEE Intl. Workshop on Distributed System: Operations and Management(DSOM99). [s.l.]:[s.n.],1999.
- [8] Hwang K C, Hong J J, Lee K H. A SNMP Group Polling for the Management Traffic [C]//Proc. of the IEEE Region10 Conf. . [s.l.]:[s.n.],1999:797-800.
- [9] 刘 翔,沈明玉. 基于 SNMP 与 Web 的服务监控系统[J]. 合肥工业大学学报(自然科学版),2008(1):44-47.
- [10] Stallings W. SNMP, SNMPv2, SNMPv3 and RMON1 and 2 [M]. 3rd ed. [s.l.]:Addison-Wesley Pub Co,1998.
- [11] 张 彤,吴世荣. 基于 SNMP 计算机网络流量监控系统研究[J]. 计算机技术与发展,2011,21(1):45-48.
- [12] 吴 昊,杨 凯,胡 术,等. SNMP 跨平台移植和交换机端口 IP 探测[J]. 计算机技术与发展,2008,18(11):21-24.

(上接第 134 页)

- [11] Citro S, McGovern J, Ryan C. Conflict Management for Real-time Collaborative Editing in Mobile Replicated Architectures [C]//Proc of the 13th Australasian Confine Computer Science. [s.l.]:[s.n.],2007:115-124.
- [12] Xue Liyin, Orgun M. Locking without Requesting a Lock: A Consistency Maintenance Mechanism in Internet-based Real-time Group Editors [J]. Journal of Parallel and Distributed Computing,2005,65(7):801-814.
- [13] Liu Ling, Li Baizhan, Yang Mingyu. Research on Conflict Model for Collaborative Design in Architecture[J]. Journal of Engineering Graphics,2006,45(2):189-193.
- [14] Ma Chenhua, Lu Guodong, Qiu Jiong. Conflict detection and resolution for authorization policies in workflow systems [J]. Journal of Zhejiang University (Science A: An International Applied Physics & Engineering Journal),2009,10(8):1082-1092.