

文章编号:1000-582X(2005)05-0068-04

基于预测的图案协同设计中智能锁研究*

卢刚¹, 惠怀海², 卜佳俊², 陈纯²

(1. 杭州科技职业技术学院 计算机系, 浙江 杭州 310012; 2. 浙江大学 计算机学院, 浙江 杭州 310027)

摘要:在分布式协同设计中,为了防止并行操作出现冲突常常采用加锁机制。图案协同设计是分布式协同设计的重要应用之一,有与其相适应的特定加锁方式。提出一种新的基于预测的智能锁,这是一种系统预测锁(SFL),系统采用扩展锁定集合预测法帮助在线操作用户预测其在未来的操作区域,并提前锁定该区域,当锁定时有冲突发生,则启动相应的冲突解决策略。通过该锁机制,系统可以在一定程度上智能地协助用户提前加锁,从而保证用户实现设计的流畅操作,并预防可能发生的操作冲突。这个机制已经应用到原型系统 CoDesign。

关键词:CSCW; 协同图案设计; 扩展锁定集合; 系统预测锁

中图分类号:TP391.72

文献标识码:A

20世纪人类的杰出成果之一计算机技术把人类社会带入了信息化时代。伴随着信息化进程的不断深入,通信技术、计算机及网络技术相融合,产生了一个新的研究领域—计算机支持的协同工作(Computer Supported Cooperative Work-CSCW)。在CSCW研究和应用的众多领域^[1]中,图案协同设计是分布式协同工作的一个重要应用。随着基于网络的异地分布式、并行、协同产品开发模式的发展^[2],基于Internet的图案协同设计应运而生,它可以使位于不同地理位置的协同设计者借鉴、共享其他成员的知识和经验,对同一个任务作品进行共同操作,协同完成图案的设计和制作,从而极大地提高设计的质量和效率,当然协同设计中Internet的安全问题也刻不容缓^[3]。笔者探讨图案协同设计中的锁问题,基于图案协同设计原型系统CoDesign^[4]。

1 锁的背景

锁在图案协同设计系统中用来保持一致性,锁的使用能大大降低冲突操作发生的次数。当某一个用户试图操作一个对象/区域时,需要获得在这个对象/区域上的一个排他锁^[5]。例如,要移动一个对象,就首先要获得这个对象上的锁,这就保证了只有一个用户,即锁的拥有者操作这个对象,从而避免冲突的产生。锁按不同的标准有不同的分类,现有文献常见的分类

有强制锁与可选锁^[6-7],非立即锁与立即锁^[7],前锁与后锁^[8],对象锁与区域锁^[6-7,9],用户锁与系统锁^[10]。

笔者提出的智能锁是一种基于预测的系统锁——SFL,其主要功能是系统采用扩展锁定集合预测法帮助在线操作用户预测其在未来的操作区域,并提前锁定该区域,当锁定时有冲突发生,就启动相应的冲突解决策略。通过该锁机制,系统可以在一定程度上智能地协助用户提前加锁,从而保证用户实现设计的流畅操作,并预防可能发生的操作冲突。

2 系统预测锁

2.1 系统预测锁的定义

定义1 系统预测锁:ForecL^S为五元组。

$ForecL^S = \{ ForecL^S U_k, ForecL^S U_k^{orientation}, ForecL^S U_k^{region}, ForecCoEdL^S U_k^{region}, ForecEdL^S U_k^{region} \}$ 其中:

ForecL^S U_k 为系统预测锁的拥有者; ForecL^S U_k^{orientation} 为系统预测锁待锁定的方向; ForecL^S U_k^{region} 为系统预测锁待锁定的区域; ForecCoEdL^S U_k^{region} 为系统预测冲突后彼此竞争锁定的区域; ForecEdL^S U_k^{region} 为系统预测冲突解决后的最终锁定区域。

定义2 用户锁:L^u为二元组。

$L^u = \{ L^u U_x^{region}, L^u U_x^{time} \}$ 其中:L^u U_x^{region} 为某用户锁的区域; L^u U_x^{time} 为某用户锁定时间。

* 收稿日期:2005-01-04

基金项目:浙江省自然科学基金重点重大项目(Z603231)

作者简介:卢刚(1974-),男,浙江东阳人,杭州科技职业技术学院讲师,硕士,主要从事CAD/CAM、CSCW的研究。

2.2 系统预测锁的实现流程

1) 原始图案栅格化。系统将一个图案协同设计二维空间分割为 n 个栅格。

2) 确定某用户锁区域 $L^u U_k^{region}$ 和其系统预测锁 $ForecL^S U_k$ 。

以某用户 U_k 第 1 次点击的位置为基准, 锁定其周围的 8 个栅格, 确定用户 U_k 的锁定区域 $L^u U_k^{region}$, 同时确定了其系统预测锁的拥有者 $ForecL^S U_k$ 。为了简化说明算法, 在此仅以右下方的 $L * L$ 个单位栅格 $\{R_1, R_2, R_3, R_4\}$ 为例进行初始锁定操作, 如图 1 所示。

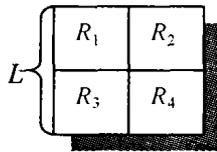


图 1 $L * L$ 个单位栅格

3) 确定系统预测锁待锁定方向 $ForecL^S U_k^{orientation}$ 。处理采集到用户 U_k 在上述 $L * L$ 个单位栅格中的操作信息, 根据计算操作强度确定预测锁待锁定方向 $ForecL^S U_k^{orientation}$ 。

① 计算各个栅格操作强度

定义 3 操作强度 I

$$I_{R_i}^k = \sum_{j=0}^n N_j^R \alpha_j, j \in [0, n], n \in [1, 2, 3 \dots]$$

$I_{R_i}^k$ 为用户 U_k 在区域 R_i 上的操作强度; N_j^R 为用户在区域 R_i 上, 时间槽 Timeslot j 上的操作次数, 以点击次数表示操作次数, 时间槽如图 2 所示。

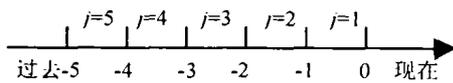


图 2 时间槽

n 由系统确定, 本系统取 5, 即采样的对象为最近的 5 个时间槽, 若在 5 个时间槽内用户没有操作则自动解锁。 α_j 为权重, 由 $\alpha_j = 2\alpha_{j+1}$ 定义, α_j 沿坐标轴逆方向线性递减, 其初始赋值为 1。

② 栅格排序

得到 $L * L$ 个栅格的操作强度后, 对这些栅格由高到低进行排序。以集合 $\{R_1, R_2, R_3, R_4\}$ 为例, 其对应操作强度为集合 $\{I_{R_1}, I_{R_2}, I_{R_3}, I_{R_4}\}$ 。操作强度由高到低排序后分别记做 $I_{1st}, I_{2nd}, I_{3rd}, I_{4th}$ 。若有关系 $I_{R_2} > I_{R_1} > I_{R_4} > I_{R_3}$, 则得到新的序列: R_2, R_1, R_4, R_3 , 分别记为: $R_{1st}, R_{2nd}, R_{3rd}, R_{4th}$ 。

③ 提取基准栅格集合

$$\text{基准栅格集合} = \begin{cases} \{R_{1st}\} & \text{if } I_{1st} \geq 2I_{2nd}; \\ \{R_{1st}, R_{2nd}\} & \text{if } I_{1st} < 2I_{2nd}. \end{cases}$$

一旦确定了基准栅格集合, 即确定了系统预测锁待锁定方向 $ForecL^S U_k^{orientation}$ 。

4) 确定系统预测锁待锁定区域 $ForecL^S U_k^{region}$ 。

确定系统预测锁待锁定区域 $ForecL^S U_k^{region}$ 即确定系统待锁定的栅格集合 W , 具体做法是以基准栅格为出发点, 将与基准栅格相邻的未被锁定的栅格加入集合 W 。如果已锁定集合 $\{R_1, R_2, R_3, R_4\}$, 则:

Case 1: 若基准栅格集合为 $\{R_{1st}\}$, 即示例中的 $\{R_2\}$, 则此时首先将与 R_2 相邻的未被锁定的 $\{R_5, R_6\}$ 加入集合 W , 如图 3 所示。

其次, 如果新加入集合 W 的栅格所夹的栅格没有被锁定, 则将这些所夹的栅格亦加入待锁定集合 W 。图 4 中 $\{R_7\}$ 为 $\{R_5, R_6\}$ 所夹的没有被锁定的栅格, 则将 $\{R_7\}$ 加入集合 W 。

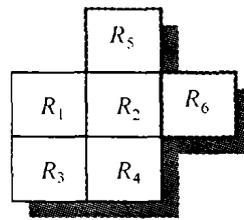


图 3 首先加入 R_5 和 R_6

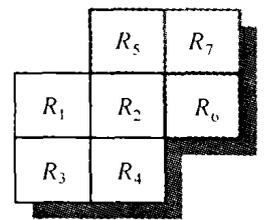


图 4 再加入 R_7

最终待锁定栅格集合 W 为 $\{R_5, R_6, R_7\}$, 将集合 W 进行加锁, 若锁定成功, 加入已锁定集合 $\{R_1, R_2, R_3, R_4\}$, 得到集合 $\{R_1, R_2, R_3, R_4, R_5, R_6, R_7\}$ 。

Case 2: 若基准栅格集合为 $\{R_{1st}, R_{2nd}\}$, 即示例中 $\{R_2, R_1\}$, 则首先将与 $\{R_2, R_1\}$ 相邻的未被锁定的 $\{R_5, R_6, R_7, R_8\}$ 加入集合 W , 如图 5 所示。

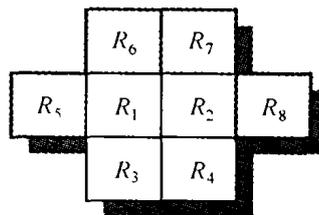


图 5 首先加入 R_5, R_6, R_7 和 R_8

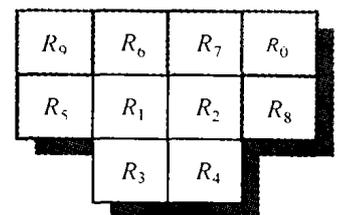


图 6 再加入 R_9 和 R_0

其次, 如果新加入待锁定集合 W 的栅格所夹的栅格没有被锁定, 则将这些所夹的栅格亦加入待锁定集合 W 。图 6 中 $\{R_9\}$ 为 $\{R_5, R_6\}$ 所夹的没有被锁定的栅格, 则将 $\{R_9\}$ 加入集合 W , $\{R_0\}$ 为 $\{R_7, R_8\}$ 所夹的没有被锁定的栅格, 则将 $\{R_0\}$ 加入集合 W 。

最终待锁定栅格集合 W 为 $\{R_5, R_6, R_7, R_8, R_9, R_0\}$, 将集合 W 进行加锁, 若锁定成功, 加入已锁定集合 $\{R_1, R_2, R_3, R_4\}$, 得到锁定集合 $\{R_1, R_2, R_3, R_4, R_5, R_6, R_7, R_8, R_9, R_0\}$ 。

3 系统预测锁的冲突解决

3.1 冲突类型

运用系统预测锁时会有两种冲突情况: 一是系统要为 2 个以上用户对同一区域进行加锁时, 即在 $ForecL^S U_1^{region}, ForecL^S U_2^{region} \dots$ 之间存在冲突的可能性; 二是系统要为一个用户对一个已经被其它用户加

锁区域进行预测加锁时,即 $ForecL^S U_k^{region}$ 和 $L^u U_k^{region}$ 存在冲突的可能性。如图 7 ~ 8 所示。

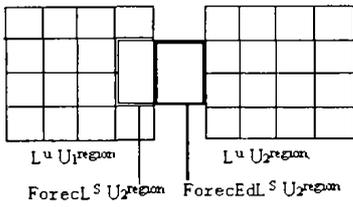


图7 系统预测锁2和用户锁1冲突示意图

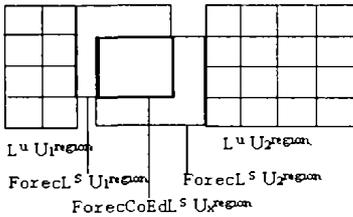


图8 系统预测锁1和2冲突示意图

3.2 冲突处理流程

```

if (t = 10) // 如果冲突超过 10 s, t 为试验值。
{
  if (ForecLS Ukregion ∩ Lu Uxregion ≠ ( ) then
  /* 系统预测加锁区域和某一用户已加锁区域冲突 */
  Compete(Op) -> Add(Op) /* 加入竞争操作, 准备解决冲突 */
  CompeteProt1 /* 选择竞争协议, 解决冲突 */
  ForecEdLS Ukregion /* 系统预测冲突解决后的最终锁定区域 */
  end if
  if (ForecLS U1region ∩ ForecLS U2region ≠ ( ) then
  /* 系统预测加锁区域彼此冲突 */
  Compete(Op) -> Add(Op) /* 加入竞争操作, 准备解决冲突 */
  ForecCoEdLS Ukregion /* 判断系统预测冲突后彼此竞争锁定的区域 */
  CompeteProt1 /* 选择竞争协议, 解决冲突 */
  ForecEdLS Ukregion /* 系统预测冲突解决后的最终锁定区域 */
  end if
}

```

3.3 处理冲突的竞争协议

系统预测锁冲突问题主要由竞争协议进行解决, 此协议定义如下:

定义4 竞争协议 CompeteProt1 为五元组。

CompeteProt1 = { Compete(Op), Protocol 1, Protocol 2, Protocol 3, Protocol 4 }, 其中:

Compete(Op): 竞争操作指针, 当有冲突发生时指针指向冲突解决协议。

Protocol 1: Popedom(L^u) (Popedom($ForecL^S$), 用户锁的优先权高于系统预测锁。即不能因为系统的预测功能干扰其它用户的正常的设计行为, 系统预测锁锁定的方向、区域仅限于没有其它用户加锁的“空白”区域。

Protocol 2: Popedom($ForecL^S U_m^{region}$) (Popedom ($ForecL^S U_{region_n}$), 系统根据用户角色分工不同有自定义的优先权。例如定义用户 M 的优先权高于用户 N , 则此时用户 M 的系统预测锁定优先权也相应高于用户 N 的系统预测锁的优先权。

Protocol 3: Popedom($ForecL^S U_k^{region}$) (Popedom ($ForecL^S U_{k+1}$), First in First out - FIFO 机制。如果某用户 X 为先进入设计空间的用户, 则它的系统预测锁优先权高于其后来者 $X + 1$ 。

Protocol 4: Popedom(Protocol 1) > Popedom (Protocol 2) > Popedom (Protocol 3), 如果有混合和其它冲突发生, 或者上述协议之间冲突, 则协议 1 的优先权高于协议 2, 协议 2 的优先权高于协议 3。

4 智能锁的系统实现

4.1 实现图示

因为该锁机制的许多步骤实现都是由系统根据算法和协议以隐式方式完成, 所以无法以全境式地图示方式表达。图9、图10列出了系统预测锁实现过程的主要显式步骤。

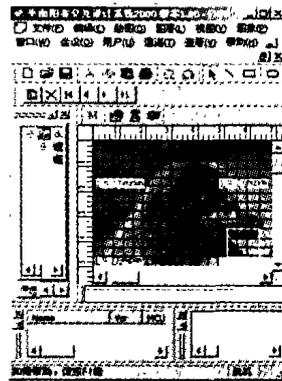


图9 用户加锁情况

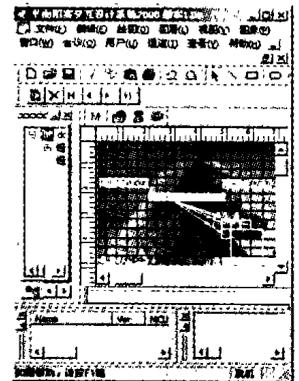


图10 系统预测锁为用户预测加锁

4.2 系统预测锁的优点

1) 智能性。系统对用户自动进行预测加锁, 并在一定的时间段自动解锁, 且可以让用户自由选择采用系统预测锁与否, 使系统预测锁具有一定的智能性。

2) 实用性。系统预测锁可以让用户开始操作时放心地选择锁定合适恰当地操作区域, 经过反复试验证明有很好的实用性。

由于用户知道有系统预测锁帮助其锁定未来操作区域, 因此他可以在开始设计时放心地锁定一个合适的区域即可, 而不是贪图一次性锁定一个较大的区域。这样既可以避免因为一个用户提前锁定一个较大的区

域(大部分闲置)而影响其它用户的操作,也不会因为未来想要操作的区域被别的用户锁定而影响自己操作的流畅性。同时还保证了后加入用户仍可以开始新的设计操作,不至于由于其它在线用户的浪费“锁定”而致使新用户没有操作区域或者操作区域过小。

3) 预防性。由于系统预测锁是前锁机制,因此可以有有效的预防冲突的发生,并且大多数冲突可以由系统锁以隐式的方式解决,从而大大减少用户因冲突而造成设计时间的浪费。

5 结 语

加锁的目的是为了更好的进行协同设计,所以加锁必须从方便用户的角度出发,充分考虑操作的方便性和实用性,并有效的降低用户锁定的工作量。

协同图案设计在纺织印染、机械加工,以及建筑等领域有广泛的前景,然而传统的锁机制显得过于单调并缺乏智能性。针对此问题笔者提出了基于预测的智能锁机制,由系统采用扩展锁定集合预测法帮助在线操作用户预测其在未来的操作区域,并提前锁定该区域,当然如果在锁定时有冲突发生,就启动相应的冲突解决策略。通过该锁机制,系统在一定程度上实现了智能地协助用户提前加锁,有效地保证了用户设计的流畅操作,同时预防可能发生的操作冲突。

感谢武频博士后、徐向华博士、黎勇博士和纪培蒂硕士对本研究的支持,感谢 Mr. Terry 阅读全文,并提出的建议。

参考文献:

- [1] 史美林,向勇. 计算机支持的协同工作理论与应用[M]. 北京:电子工业出版社,2001.4-7.
- [2] 尹超,郭坤毅. 网络化协同产品开发的决策框架模型及其应

- 用[J]. 重庆大学学报(自然科学版),2004,27(6):5-9.
- [3] 叶芳,吴中福. 网络入侵的聚类算法研究与实现[J]. 重庆大学学报(自然科学版),2004,27(3):46-52.
- [4] JIANG BO, CHEN CHUN, BU JIAJUN. CoDesign - A Collaborative Pattern Design System Based on Agent[A]. Proceedings of the 6th International Conference on CSCW in Design[C]. Ottawa: NRC Research Press, 2001. 319-323.
- [5] CHANG C T K. Adaptive Multi-granularity Locking Protocol in Object-oriented Databases[Z]. PhD Dissertation, Department of Computer Science, University of Illinois at Urbana-Champaign, 2002.
- [6] SUN CHENG-ZHENG. Optional and Responsive Fine-grain Locking in Internet-based Collaborative Systems[A]. Proceedings of the IEEE Transactions on Parallel and Distributed Systems[C]. HongKong: IEEE Computer Society Press, 2002, 13(9): 65-72.
- [7] CHEN DAVID, SUN CHENG-ZHENG. Optional and Responsive Locking in Distributed Collaborative Object Graphics Editing Systems[A]. Proceedings of ACM SIGGROUP[C]. New York: ACM Press, 1999, 20(3): 17-20.
- [8] XUE L, ZHANG K, SUN C. Conflict Control Locking in Distributed Cooperative Graphics Editors[A]. Proceedings of the 1st International Conference on Web Information Systems Engineering (WISE 2000)[C]. HongKong: IEEE Computer Society Press, 2000. 401-408.
- [9] NEWMAN-WOLFE R E, WEBB M L, MONTES M. Implicit Locking in the Ensemble Concurrent Object-oriented Graphics Editor[A]. Proceedings of ACM on CSCW 92[C]. Toronto: ACM Press, November 1992. 265-272.
- [10] XU XIANG-HUA, CHEN CHUN, BU JIA-JUN. Distributed Dynamic-locking in Real-time Collaborative Editing Systems[A]. Proceeding of 10th international conference on groupware (CRIWG2004)[C]. Costa Rica: IEEE Computer Society Press, September 2004. 165-172.

Intelligent Locking in Collaborative Pattern Design Based on Forecast

LU Gang¹, HUI Huai-ha², BU Jia-jun², CHEN Chun²

(1. Department of Computer, Hangzhou Poly Technique College, Hangzhou 310012, China;
2. College of Computer Science, Zhejiang University, Hangzhou 310027, China)

Abstract: Locking is a common technique in distributed collaborative pattern design systems, which is used to prevent operation conflicts, maintain user intention and result consistency. There is a special locking scheme in Collaborative Pattern Design system, a special class. A novel scheme System Forecast Lock (SFL) is proposed, so that some regions are forecasted by system where user wants to operate in the future, and then these regions are added into locked set. Strategy is used when conflicts happen. By using the scheme proposed, system locks regions intelligently for the sake of user convenience, ensures user's fluency of operate, and prevents operational conflict. The proposed scheme has been implemented in CoDesign system.

Key words: CSCW; collaborative pattern design; extend locked set; system forecast lock