Maintaining Semantic Consistency in Real-Time Collaborative Graphics Editing Systems

Jiajun Bu[1], *Bo Jiang*[2] *and Chun Chen*[1]

[1]College of Computer Science, Zhejiang University, Hangzhou, P.R.China
[2]College of Computer Science and Information Engineering,
Zhejiang Gongshang University, Hangzhou, P.R.China

Summary
To make the real-time collaborative graphics design system more efficient, it is essential that the semantic violation problem be well solved. However, works on semantic preservation are very limited. We present the novel schema of semantic locking to prevent semantic violation. Semantic locks are classified into region lock and object lock to solve semantic violation problems at different level of granularity. Users' intension can be expressed either by attaching comment to the lock explicitly or by making some rules to the lock implicitly. A new approach that can resolve semantic conflict problem efficiently is proposed in this paper. The schema has been tested in the CoDesign system.

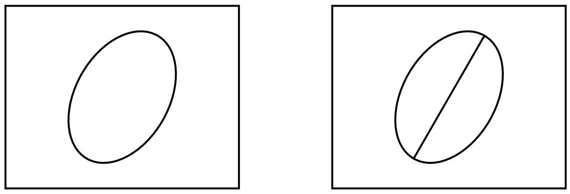
Key words:
CSCW, real-time collaborative graphics design, semantic preservation, semantic expression.

Introduction

Internet-based collaborative graphics design systems are a special class of Computer-Supported Collaborative Work (CSCW) system [1, 2, 3, 4], which allow several users geographically distributed to view and design same graphics document simultaneously. Distributed graphics editing systems can be further divided into two categories: image-based and object-based. In this paper, we are particularly interested in real-time graphics design systems that are object-based.

Real-time, distributed, and unconstrained are three basic characteristics of these systems [5]. To achieve high responsiveness, a replicated architecture is adopted. Operations are executed locally and replicated to other sites. Intension preservation problem caused by replicated architecture can be classified into two major problems: syntactic preservation and semantic preservation. Although syntactic preservation problem has been well resolved [6, 7, 8], little has been done in solving the semantic preservation problem. Without semantic preservation collaborative design will be less efficient for users cannot well understand others' meaning.

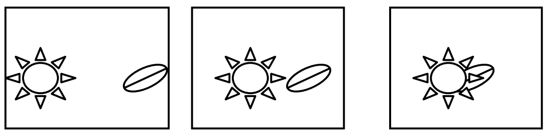
Considered the following situation, the intension of user A is to draw a green leaf as shown in Fig. 1. After user A draws an ellipse, he does something else before goes back to continue designing the leaf. User B fills it red to design a red balloon in the region during the interval.



(a) Initial Drawing (b) The intension of user A is to draw a leaf.

Fig. 1. Example1 of semantic violation.

Due to concurrent generation of operations, the actual effect of an operation at the time of its execution may be different from the intended effect of this operation at the time of its generation. As shown in Fig. 2- (1) (2), both user A and user B deem that the leaf should be one part of the plant and cling to the flower. User A moves the flower right and at the same time user B moves the leaf left. Although two users' intension is consistent, the concurrent operations lead to semantic violation problem. Fig.2- (3) shows that two objects are overlapped.



(1) Initial Drawing (2) Intension of Two Users (3) Final Drawing

Fig. 2. Example2 of semantic violation.

In this paper, we analyze semantic preservation problem and present both region semantic locking and object semantic locking to settle the semantic preservation problem in collaborative graphics design systems that are object-based. Besides, an approach to resolve semantic

Manuscript received March 5, 2006.
Manuscript revised March 30 , 2006.

conflict problem is proposed. The goal of the schema is to make collaborative design more efficiently and smoothly.
The rest of this paper is structured as follows. Section two analyzes the problem of semantic violation. Section three presents the schema of semantic locking in CoDesign. Section four proposes semantic conflict resolution. Section five compares our approach to the related work. Finally, Section six concludes the paper.

2. Semantic Preservation

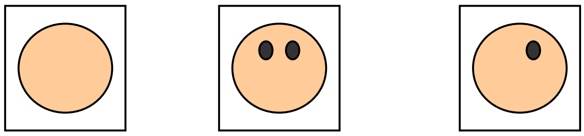
Syntactic preservation, which is widely used in real-time collaborative graphics designing system, aims at promising the same operation execution order and result at all collaborative sites, so that each user's operations can be expressed orderly and consistently to other users. But user's meanings may not be understood with each other only through viewing the operation execution, not to say that users' operations always interleave with each other and difficult to distinguish. Because the departure of user's meanings may cause confusion and low efficiency in the collaborative work, additional measures should be adopted if we want to share these meanings in the collaborative group.
In collaborative text editing system, the problem is not serious. User's meanings can easily be understood with each other when all users comply with certain language's syntax. That is to say, semantic preservation can be achieved by complying language syntax in collaborative text editing system. But that's not the case in the collaborative graphics designing system. It is almost impossible to constitute complete designing rules and demand users to comply them.
Now we analyze the semantic preservation problem in detail. We classify this problem into two categories: static semantic preservation and dynamic semantic preservation.

2.1 Static Semantic

Objects are created and modified with user's meanings. For example, if a user wants to draw a face, maybe he draws two circle objects as eyes, a curve object as mouth, etc., so all objects are created with user's meanings (i.e. facial feature). When these meanings are preserved, we call it static semantic preservation. Since these meanings are related with one object or many objects, we can further classify the static semantic preservation into single object semantic preservation and object-object semantic preservation:

• Single object semantic preservation

Single object, such as line, circle, contains user's meanings. These meanings can be expressed as object attributes, such as color, size, etc. For example, as shown in Fig. 3a, user A draws a rectangle. If user B modifies the color of that circle, user A's meaning is violated (Fig. 3c). We call this type of semantic preservation situations as single object semantic preservation. If user does not want other users to modify these attributes (i.e. meaning) arbitrarily, measure should be adopted to keep them and notify other users, and other users will operate the object with certain consideration.



(a)Initial .

(b) User A adds two black circles as eyes.

(c)User B erases a circle, and violates user A's semantic intention.

Fig. 3 Example of semantic violation

Unlike the object lock, in single object semantic preservation, users only restrict operating some attributes of the object but not prohibit operating the whole object. For example, if user A fixes the small circles' color (black color denotes eyes) in Fig. 3b, other users can still change position of the circles without violating user A's meaning. Furthermore, the restrictions on object are not imperative, and other users can still modify these restrictions through certain mechanism.

• Object-object semantic preservation

There are also containing user's meanings among objects. These meaning can be expressed as object relations, such as relative position, overlap order, etc. For example, as shown in Fig.3b, user A regards the big yellow circle as a blank face, and two small black circles as eyes. Then the relative position of the three circles contains user A's meaning (i.e. a face with only eyes on it). We call this type of semantic preservation situations as object-object semantic preservation. These meanings can also be preserved and notified to other users if user A wants to, and then other users cannot modify the related objects arbitrarily.
Note that object-object semantic preservation is unlike group operation. Objects are grouped and changed the same attributes in the group operation. After the operation is finished, these objects are ungrouped. In object-object semantic preservation, the restrictions on the objects are always available, and the unrestricted attributes of these objects can still be modified freely. For example, if user A fixes the position and layer attributes of the three circles in Fig. 3b, users can change the color of any circle without violating the meanings of user A.

2.2 Dynamic Semantic

Except static semantic preservation, there also exists dynamic semantic preservation problem. For example, as shown in Fig. 1a, user A regards the big yellow circle as a blank face, and intends to draw facial features on it. If he does not want other users to break it, his meanings on the object should be preserved and notified to other users. Another example, if user A wants to draw a flower picture and do not want to use the black or gray color on it, these meanings should also be preserved and notified to other users. We call this type of semantic preservation situations as dynamic semantic preservation. It means user plans to do something. Measure should also be adopted to express these meanings to other users.

In above semantic preservation problems, user's meanings are only preserved when he intends to preserve them, but not that these meanings must be preserved. The static semantic preservation focuses on preserving the existent meanings contained in the objects, and the dynamic semantic preservation focuses on preserving user's purpose. Both of them are useful if we want to achieve reciprocal understanding of all users, reduce semantic conflict, and enhance efficiency of collaborative work.

3. Semantic Locking Scheme

In CoDesign, locking can be applied to drawing regions or graphical objects to prevent semantic violation from occurring on them. Semantic lock can be expressed as follows:

SL:= (SV, Position of the region/ObjID, <[Comment] , [(M_Attrib1 [, Value Range]), [(M_Attrib2 [, Value Range])],…,| [NM_Attrib1], [NM_Attrib2]],…>)

SV denotes the state vector [5], which is used to record the time that user locks the region or object. The locking area can be defined with a rectangle. ObjId denotes the identification of the object or object group, which is unique in the pattern document. Comment denotes the description that user want to explain. M_Attrib denotes the attribute (such as color, position, and etc.) that may be modified by others, whereas NM_Attrib denotes the attribute that may not be modified by others. These two kinds of attributes are selective. Value denotes the attribute value. As the square brackets denote the elements in it are optional, user can express his meaning by the comment or the definition of the attributes. While a region is locked, all drawing objects in the region are locked. Therefore, if the expression of a region lock includes the items of M_Attrib or NM_Attrib, all the objects in the region should also comply with the rules.

To preserve user A's meaning in Fig.1., the object lock can be expressed as follows:
SL1:=(SV, Obj1, "leaf")
SL2:=(SV, Obj1, "leaf", (COLOR, GREEN))

The following expression of region lock can be applied to prevent semantic violation in Fig. 2.
SL3:=(SV, ((1, 1), 100, 120)), | POSITION)
SL4:=(SV, ((1, 1), 100, 120)), "flower clings to leaf", | POSITION)

Semantic preservation is achieved by attaching intension comments to the lock explicitly or by placing rules on confined object lock implicitly.

After the semantic locking is issued, it is transmitted to other sites. When user selects and checks the objects, the semantic expression related to the objects will be shown in the window. If user violates the rules, warning message will be popped up and the operation will be undone.

Semantic Lock can be classified into two types: shared and confined. The expression of a shared lock only includes comment. Users prevent violating others' intension on one's own initiative by viewing the semantic comment related to the object or region shown in the window. The creator of the confined lock has full rights on it. However, other users can only issue the permitted operations on the objects or the region.

4. Semantic Conflict Solution

In replicated architecture, concurrent operations may lead to conflict. Like drawing operations, semantic locking operations can also cause conflicting problem.

As shared lock expresses user's meaning by attaching comment that is ambiguous, we focus on presenting the resolution that related to confined lock in this paper. Though semantic locks are separated as region lock and object lock, the rules regulated the region also constrain the objects within the region. Therefore, semantic conflict that object-based will be the key issue.

In CoDesign, operations can be classified into two categories: traditional editing operation (EO) and semantic locking operation (SLO).

The semantic operation relations can be conflicting or compatible. The definition of conflicting relation and compatible relation are given as follows:

Definition 1. Semantic conflicting relation "⊗"

Given two operations O1 and O2 generated from site i and j each, at lease one operation is SLO, if and only if: (1) O1 and O2 are concurrent, (2) O1 is EO or SLO, and O2 is SLO, O1 operates on the attribute of the same object that O2 not permitted or with different values.

Definition 2. Semantic compatible relation "©"

Given two operations O1 and O2 generated from site i and j each, at lease one operation is SLO, if and only if O1 and O2 are not conflicting.

If the conflict involves both SLO and EO, then only SLO is applied to the object. As shown in Fig. 2-(1), user A issues SL3 and user B sends the EO1 to move the leaf left simultaneously. When EO1 conflicts with SL3, we preserve SL3 and remove EO1. User A moves the flower right subsequently. Thus, two users' intension is preserved. It is also possible that O1⊗O2 and two operations are SLO. Moreover, in a highly concurrent real-time collaborative editing environment, a group of operations may have rather arbitrary and complex semantic conflict relationships among them. For example, O1 and O2, O2 and O3 are conflicting, but O2 and O3 are compatible. A distributed multi-version approach is proposed to settle the problem.

Given a group of n operations, O1, O2, …, On, targeting the same object, their semantic conflict relationship can be fully and uniquely expressed by a n × n matrix, in which element M[i, j], 1≤i, j≤n is filled with "⊗" and "©". For example, 3×3 matrix for four operations is shown in Fig. 4.

Relation	O_1	O_2	O_3
O_1		⊗	©
O_2	⊗		⊗
O3	©	⊗	

Fig. 4. Relation between three operations.

Algorithm 1: Given a matrix of a group of N concurrent SLOs. Semantic Compatible Group Sets (SCGS) can be expressed as follows:

SCGS = {SCG1, SCG2, …, SCGn}

All operations in a single SCGk are mutually compatible.

1. SCGS={};
2. For 1 ≤ i ≤ N, and i < j ≤ N
   If M[i, j-1]= © and Oi or Oj is SLO
   Then SCGS=SCGS+{{ Oi , Oj }}
3. For 1 ≤ i ≤ N
   If Oi is SLO, and not in SCGk,
   1≤k≤|SCGS|
   Then SCGS=SCGS ∪ {{Oi}}.

After the SCGS is formed, users will negotiate to choose one SCG to apply on the object if there are more than one SCG in the SCGS.

5. Comparison to Related Work

Systems like TeamWorkStation[9] and VideoDraw[10] applied video channels to make the users keep in touch in their collaborative works. But the systems can only support 2-3 people to work simultaneously and users' meaning may not be understood clearly only by observing the video.

Some work have been done to solve the semantic reservation problem in CoDesign[11]. In the former model, it is compulsory to insert comments in semantic operations to express user's intension and the semantic operations are only object-based. These may impose fussy work to users. Moreover, the problem exists in Example 2 in this paper cannot be solved.

6. Conclusion

In this paper, we analyze the semantic preservation problem and propose semantic locking to resolve the semantic violation problem. Main contributions of this paper include the presentation of semantic locking at different level of granularity, multiple expressions of semantic locking (comments or rules), and the multi-version and negotiation approach to solve the semantic conflict. The semantic locking schema proposed in this paper has been implemented in the CoDesign prototype system and the schema makes collaborative design more efficiently.

Acknowledgments

This paper is supported by National Natural Science Foundation of China (60573176) and Zhejiang Provincial Natural Science Foundation of China under Grant No. Z603231.

References

[1] Bo Jiang, Chun Chen, Jiajun Bu, "CoDesign-A collaborative pattern design system based on agent", Proceedings of the Sixth International Conference on Computer Supported Cooperative Work in Design, Canada, p319-323, 2001.7.

[2] R.E. Newman-Wolfe et al., "MACE: A Fine Grained Concurrent Editor," Proc. ACM COCS Conf. Organizational Computing Systems, pp. 240-254. 1991.

[3] M. Ressel, D. Nitsche-Ruhland, and R. Gunzenbauser, " An Integrating, Transformation-Oriented Approachto Concurrency Control and Undo in Group Editors", Proc. ACM Conf. Computer Supported CooperativeWork, pp 288-297, Nov. 1996.

[4] C. Sun and D. Chen, "Consistency Maintenance in Real-Time Collaborative Graphics Editing Systems", ACM Transactions on Computer-Human Interaction, 2002. 9(1): p. 1-41, 2002. 9.

[5] C. Sun, X. Jia, Y. Zhang, Y. Yang, and D. Chen, " Achieving Convergence, Causality-Preservation, and Intention-Preservation in Real-Time Cooperative Editing Systems," ACM Trans. Computer-Human Interaction, vol. 5, no. 1, pp. 63-108, Mar. 1998.

[6] D. Chen and C. Sun, "A distributed algorithm for graphic object replication in real-time group editors", Proceedings of ACM Conference on Supporting Group Work, Phoenix, Arizona, USA, pp. 121-130, Nov. 1999.

[7] T. P. Moran, et al. “Some design principles for sharing in Tivoli, a whiteboard meeting-support tool”, Groupware for Real-time Drawing: A Designer's guide, pages 24-36. McGraw-Hill, 1995.

[8] R. Kanawati, “LICRA: A replicated-data management algorithm for distributed synchronous groupware application”, Parallel computing, 22:1733--1746, 1997.

[9] Ishii, H. “TeamWorkStation: towards a seamless shared workspace”, in Proc. CSCW ’90 Conference on Computer-Supported Cooperative Work (Los Angeles, CA., October 7-10). ACM, New York, pp. 13-26, 1990.

[10] Tang, J,C. & Minneman, S.L. “VideoDraw: a video interface for collaborative drawing”, in Proc. CHZ ’90 Human Factors in Computing Systems (Seattle, WA., April 1-5). ACM, New York, pp. 313- 320, 1990.

[11] Wang, X., Bu, J., and Chen, C., “Semantic Preservation in Real-time Collaborative Graphics Designing Systems,” the 4th Intl. Workshop on Collaborative Editing. New Orleans, Louisiana, USA. Nov. 2002.

Jiajun Bu received the PhD degrees in Computer Science from Zhejiang University in 2000. He is now an associate professor in College of Computer Science of Zhejiang University. His research interests include CSCW, Ubiquitous Computing, and Computer Vision.

Bo Jiang received the M.S. degrees in Computer Science from Zhejiang University in 2002. She stayed in Ubiquitous Computing Research Laboratory (UCRL) Zhejiang University to study CSCW, and Ubiquitous Computing since 2000. She is now an associate professor in Zhejiang Gongshang University and a PhD candidate in Zhejiang University.

Chun Chen is now a professor in College of Computer Science of Zhejiang University. His research interests include CSCW, Ubiquitous Computing, Computer Vision, and CAD/CAM.