Conflict Analysis in Replicated Collaborative Solid Modeling Systems

Shuxu Jing Fazhi He Huajun Liu Bin Liao

School of Computer Science and Technology, Wuhan University, Wuhan, China, 430072

Abstract *Conflict problem is one of the major issues in Collaborative CAD systems. Based on a flexible multi-user interaction framework for replicated Collaborative Solid Modeling (CSM) systems, typical conflict issues are researched. The casualty conflict was discussed at first. Then, the name conflict was analyzed. Finally, compatibility checking for operations that may result in possible conflicts is studied. Based the analysis, some interesting results are discovered. The results will be helpful to utilize the potential advantages of replicated CSM systems in future.*

Keywords: Collaborative Solid Modeling, replicated, conflict.

1 Introduction

Computer-Supported Cooperative Work (CSCW) reflects a trend from using computer to solve computation problems to using computers to support human-to-human interaction. From (Human-Computer Interaction) HCI to HHI (Human-human Interaction) is a major development from traditional single-user CAD systems to simultaneous collaborative CAD systems. A natural, fast and less constrained interaction among a group of users is very important to simultaneous collaborative CAD systems.

As a typical simultaneous collaborative CAD system, Collaborative Solid Modeling (CSM) system involves geographically distributed users who create, view, and modify geometric model in a shared workspace. The architecture of CSM systems can be centralized or replicated [1][2][3]. The replicated CSM has some potential advantages, such as fast response, high concurrency and light network load. But the concurrent control and consistency maintenance in replicated CSM systems are more complicated than in centralized ones.

One major problem of collaborative modeling systems is that there exist various types of conflicts. R. Bidarra [4] and X. Zhou [5] discussed the conflict problem in Web-base (centralized) CSM systems. Both of them focused on geometric conflict problems.

This paper discusses several other conflict problems specific to replicated CSM system in the full-distributed environment.

2 Related works

2.1 Concurrent control methods

Typical concurrency control methods adopted in replicated CSM systems are quite strict. For example, floor control applied in TOBACO, [6] pessimistic object lock mechanism applied in Cooperative ARCADE system [7] and Public rules & protocols applied in CSCW-Feature [8], where only one user (the current floor / lock holder) can execute modeling commands and the operations on the shared model from other users is denied in order to avoid possible conflicts.

However, the strict concurrent control methods prevent simultaneous access to shared CAD data. Furthermore, the consumption of lock/unlock operations or floor control operations is also tedious during the process of interaction.

2.2 Persistent naming problems

Related works of persistent naming falls into three catalogues:

- Persistent naming of standalone parametric CAD systems. [9-16]
- Name corresponding of parametric model across heterogeneous CAD systems. [17]
- Persistent naming in centralized CSM systems [5] and web-based CSM systems. [4] [18] [19]

2.2.1 Standalone naming problem

Figure 1 is the architecture of standalone persistent naming mechanism, which is summarized and criticized by R. Bidarra, who urges to break the cyclical referencing between unevaluated model and B-rep model. After breaking, only persistent entities in the unevaluated model can be used. [14] [15]

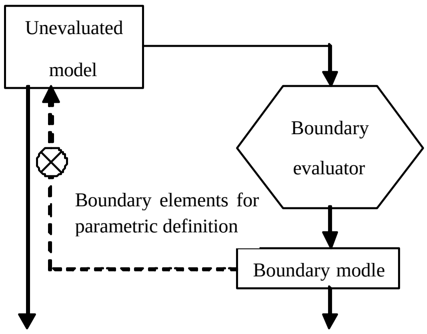

Figure 1 Standalone persistent naming problem

The key issues of standalone naming mechanism include:

- How to name topological entities in the B-rep model.
- How to uniquely recording a referenced topological entity in the parametric model.
- How to establish the correspondence of the topological entities in the re-evaluated B-rep model with their recorded names in parametric model.

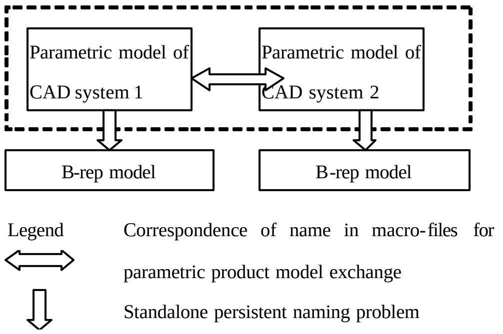

Figure 2 Relationship of two naming problems

2.2.2 Name problem of parametric product model exchange

The research work conducted in KAIST aims to exchange parametric product model across heterogamous CAD systems [17]. The persistent naming mechanisms of heterogeneous CAD systems are different, so one of the major tasks of parametric model exchange is to establish the correspondence of names involved in parametric models across heterogeneous CAD systems.

The relationship of standalone persistent naming problem and the naming problem of parametric product model exchange is illustrated in Figure 2. The history-based macro-file can be seen as parametric model [17].

2.2.3 Name problemin centralized CSMsystems

In centralized (for examples, the web-based) CSM systems [4] [5] [18] [19], there is a central modeling server, which executes all the modeling commands. The order of modeling commands is preserved by the centralized modeling server. Then the traditional persistent naming mechanisms can be applied in centralize CSM systems with slight adjustment.

Current Collaborative CAD systems, both centralized ones and Web-based ones, due to the concurrent control mechanism adopted, don't support natural and free Human-Human interaction. In this paper, we present a multi-user interaction framework for replicated CSM system, which supports natural and less constrained Human-Human interaction. Conflict problems involved in replicated CSM are analyzed.

3 A multi-user concurrent interaction method for replicated CSM systems

3.1 Replicated CSM systems

There are two types of modeling operations should be defined in replicated CSM systems. Given any site:

- A local operation is a modeling command generated by this site;
- A remote operation is a modeling command generated by any other site.

The general characteristics of replicated CSM systems can be summarized as:

- A set of participant sites connected by a communication network.
- Operations are simultaneously issued and executed immediately at local sites to obtain a fast response.
- Local operations are sent to all the other sites and they are executed remotely with transformed form.
- It is the operation events not the updated geometric data that are sent to remote sites. Thus the network traffic is reduced.
- The consistency of B-rep model at all modeling sites is achieved by exchanging of modeling commands

There always exists network latency between the operation execution at local site and the moment of its arrival at remote sites. So, the operations are executed in different orders at different sites, which will lead to conflict problems in replicated CSM systems.

3.2 Multi-user interaction method

Figure 3shows a multi-user interaction method for replicated CSM systems. The concurrent interaction and control among all the sites are processed as following steps:

- Firstly, causal relationships of received operations are preserved.

- The commutable operations are executed.
- Operation transformation is applied to the non-commutable operations so that they can be executed according to the received order but in transformed form.
- However, in solid modeling application, not all of the non-commutable operations can be transformed. A readjustment to the received order is applied.
- In cases where some types of conflicts could not be solved by order readjustments multi-version awareness method is applied.
- After a convergence result has been achieved, whether the result is a valid and meaningful one will be checked according semantic analysis.

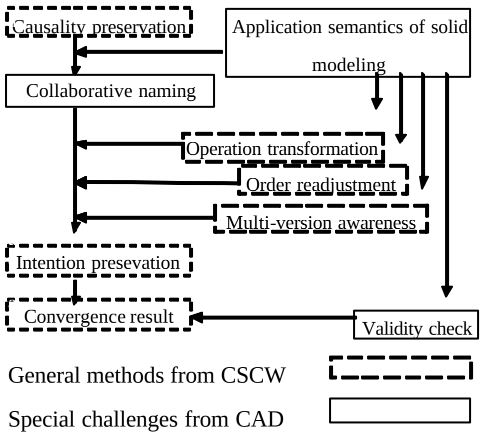

Figure 3 A multi -user interaction method

This method combines the general methods in CSCW [21] and the special challenges from solid modeling field.

4 Causal conflict analysis

StateVector have been widely used in CSCW field in replicated co-editor systems to support natural multi-user interaction [21]. This paper applies *StateVector* to analyze conflict problem in replicated CSM systems.

4.1 State vector

StateVector (SV) is a kernel concept in full-distributed computer system to describe the partial global time information across sites [22].

SV is used to capture the causal relationship among all operations in the system. Let N be the number of collaborative sites in the system. Assume that sites are identified by integers 0 to N-1. Each site maintains an SV with N components.

Initially, SV[i] = 0 for all i belongs to {0, ..., N-1}. After executing an operation generated by site i, SV[i] = SV[i]+1. An operation is executed at the local site immediately and then broadcasted to remote sites with a timestamp of the current SV.

4.2 Causal conflict solution based on SV

The causal conflict in replicated CSM systems can be illustrated as shown in Figure 4:

Assuming there are three collaborative sites (site 1, site 2 and site 3) and three operations:

- O_1 creating a block, associated with a *SV* (1, 0, 0) when it is sent to other sites.
- O_2 creating a round slot on the block, associated with a *SV* (2, 0, 0) when it is sent to other sites.
- O_3 blending the interacting edge of the block, associated with a *SV* (3, 0, 0) when it is sent to other sites.

At site 1, the three operations are generated and executed immediately in the generating order. Then they are sent to site 2 and site 3 for remote execution. Due to network latency, these three operations may be received in different order at different sites.

At site 2, operation O_1, O_2 and O_3 are received in the generating order and can be executed immediately and

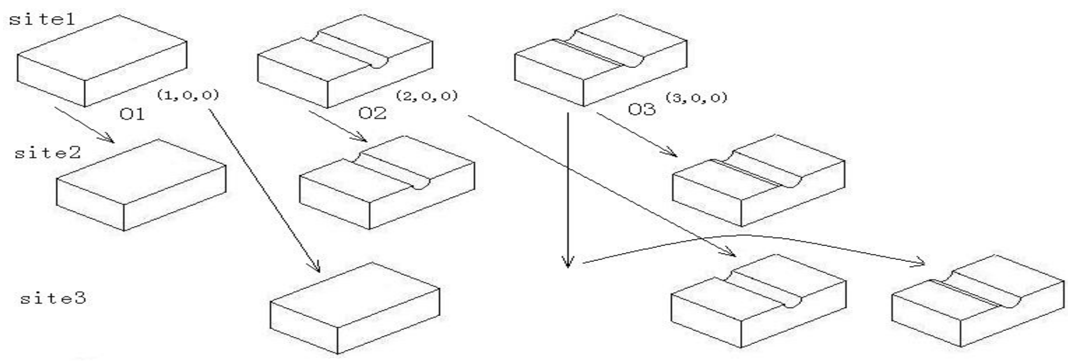

Figure 4 Causal problem of replicated CSM systems

correctly.

But at site 3, O_3 (3, 0, 0) arrived before O_2 (2, 0, 0). The edge to be blend has not been generated because O_2 has not been executed. Then O_3 has to wait until O_2 is executed so the edge to be blend is generated.

Whether and how long an operation should wait can be decided by the causal relation defined by SV. Therefore, applying SV in replicated CSM systems can solve causal conflict.

5 Name conflict analysis

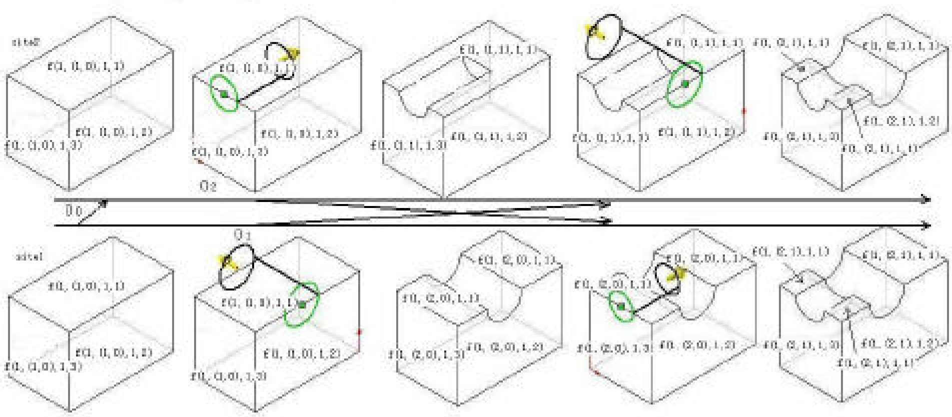

Figure 5 A case study of collaborative naming problem

Another type of conflict in replicated CSM systems is the name conflict results from the disorder execution of operations at different sites.

5.1 Collaborative naming method

There are two types of basic naming mechanisms in standalone CAD systems which can be migrated to replicated CSM systems.

1) Adopting an auxiliary data structure similar to J. Kripac's [11] method but making adjustments to name topological faces in replicated CSM systems.
2) Adopting an attribute propagation mechanism but also making adjustments to name topological face in replicated CSM.

Applying the first kind of naming methods in replicated CSM may result in some interesting conflict problems, which have been researched in our previous work [20].

In this section, the second kind of naming methods is applied in replicated CSM systems and will be discussed:

- The faceId is attached to the face in B-rep model as an attribute;
- And all the faces result form the splitting of the same face will inherit the attribute;
- The attribute can be edited after being inherited.

Let N be the number of sites in a replicated CSM systems. FaceId (f) = (*siteId*, *SV*, *solidId*, *faceIndex*), Where

- *siteId* is the identifier number of a site in a replicated CSM system.
- *SV* is the *StateVector* with which the modeling operation is associated.
- *solidId* is the Id of the solid generated by any given site.
- *faceIndex* is the index of a given face of the given solid identified by *solidId*.

5.2 An example of the collaborative naming method

An example of the collaborative naming method is shown in Figure 5. Any newly created solid will have a unique *solidId*. And within any type of solids, *faceIndex* of the solid can be uniquely determined. The *siteId* is determined by the site, which issued the operation that generated the solid. The *SV* is also associated with the face, which stands for operation execution states at any given site.

As shown in Figure 5, O_0 is issued by site 1. It generates a block immediately at local site, and then it is sent to site 2 and generates a block as well. All the face of the block will be attached an attribute as its *faceId*. *SV* (1, 0), which means 1 operation of site 1 and 0 operation of site 2 has been executed when the operation is broadcasted, is associated with the *faceId*. Then O_1 at site 1 and O_2 at site 2 is executed and sent to site 1.

During the modeling process, the *faceId* of any given face is inherited by all its descendents and *SV* is adjusted according to the state when the operation is broadcasted. From the example we can see that *faceId* is not unique, there are three faces have the same *faceId* f(1,(2,1),1,1) and they should be distingueshed under some circomstances the distinction of them can be achieved by applying traditional stand alone naming mechanisms.

From the example above, we can discover some characteristics of naming issues specific to replicated CSM systems compared with standalone naming mechanisms :

- In the standalone modeling systems there is a clear distinction between the modeling phase (naming & recording phase) and the re-evaluation phase (retrieving phase). But this distinction in replicated CSM system disappears.
- In replicated CSM system, due to network delay, when a modeling command arrives at other sites, the topological structure of the model might have been changed due to the execution of other modeling commands. Therefore, name problem, which occurs only at the re-evaluation phase in standalone modeling systems, has to be dealt with at the modeling phase even the replicated CSM system is a pure solid modeling system.
- Even in modeling phase of replicated CSM, the well-defined naming mechanism in standalone system will result in untraceable problems in replicated CSM system due to topological structure change.

To handle collaborative naming problem, two types of situations of referenced topological entities in modeling operations should be considered:

- Only the geometric information of refered topological entities is used, as discussed in section 5.3.
- The referenced topological entities are used as the operation target, as discussed in section 5.4.

5.3 Geometric information

The names of topological entities will be used in the modeling operations, which will be exchanged among all the collaborative modeling sites. As shown in Figure 6, a through-round hole is attached to the block at site 1. This operation refers to the top face and two edges of the block for the positioning of the hole.

The problem is that when the hole-attaching operation arrives at site 2, the top face and one of the edges the operation refers have been split due to the execution of the local through-round slot-attaching operation.

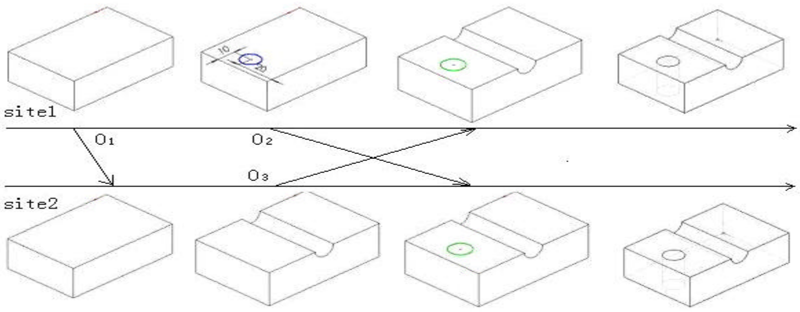

Figure 6 Only geometric information of topological entities is used

As for the positioning and orientation purpose, only geomtry of the entity is used and all the split desendents of the given face can play this role. So it is not necessary to distinguish them.

5.4 Operation target

When the referenced topological entity is used as the operation target, as illustrated in Figure 7, edge E is refered as the blending operation target.

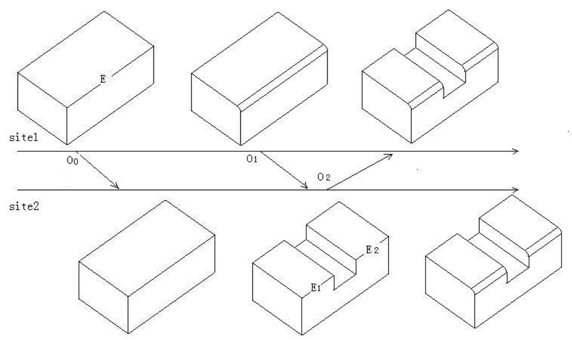

Figure 7 topological entities as operation target

O_1 blending the edge E, when it arrives at site 2 the edge E has been split by O_2, then the system must decide either E_1 or E_2 should be blended. Of course both E_1 and E_2 are blended may be an option as well. Under this situation, user interaction can be employed to make the final decision.

6 Compatibility checking for operations

There is no conflict in the example of Figure 6, so they can be considered as non-conflict operations. But in replicated CSM there may be operations that will result in conflicts.

For some types of conflict operations, operation transformation can be applied to transform them into non-conflict operations. But for some other ones, don't. Therefore, the compatibility checking should be conducted in replicated CSM.

6.1 Transformable conflict operations

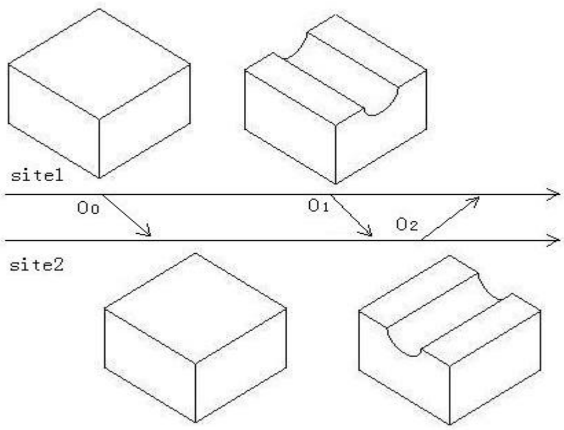

Figure 8 Transformable conflict operations

One type of conflict operation is base on different views. Take Figure 8 as an example, O_0 generates a block at site 1 and then is sent to site 2. It is executed and generates a block without question.

Then at site 1 O_1 a longitude round slot is attached to the block, at site 2 O_2 a latitude round slot is attached respectively. Either O_1 or O_2 can be tranformed because each one can be seen of same semantics if we rotate the block 90 degrees.

So we chose to execute only one of them as to achieve a consistent result. Here we don't consider the dimension value of the two slot and assume they are just the same operation.

6.2 Non-transformable conflict operations

Take Figure 9 as an example, O_0 generates a block at site1 and then is sent to site2. It is executed and generates a block without question. Then at site1 O_1 and at site2 O_2 a round through hole is attached to the block respectively.

It can be found that these two holes interact with each other. Under this situation, they are of the same type of operations but with different positioning attributes. It is not reasonable to transform either of them. Therefore, there are two candidate tragedies can be applied:

- Canceling one of them in order keep the consistent result
- Applying a multi-version method to keep both of the results.

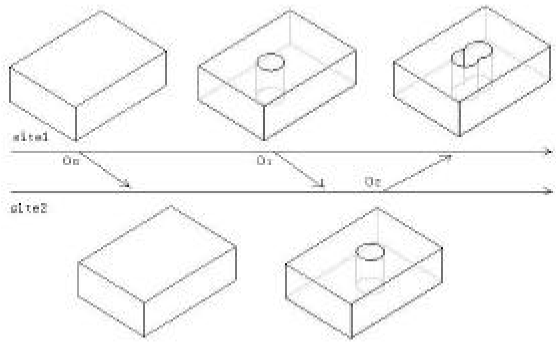

Figure 9 Non-transformable conflict operations

7 Conclusions and future works

An ideal CSM system should be high concurrency, fast response, low network traffic and natural human-human interaction, which can only be achieved in replicated CSM systems. But the conflict problems in replicated CSM systems are much more complex than in centralized ones. This paper discussed typical conflict issues in replicated CSM:

- Casualty conflict in replicated CSM systems is analyzed.
- New challenges of name problem in replicated CSM are discovered.
- Two functions of referenced topological entities are studied for the solution of name conflict.
- Transformable and non-transformable conflict operations in replicated CSM systems are illustrated with examples.

Based on the analyses in this paper, open issues in replicated CSM systems should be researched in near future.

Acknowledgements

This research has been partially supported by the NSFC foundation (on the Grant 60303029) and NSFC international cooperation project (NSFC/KOSEF 2005-2007) and Open Foundation of State Key Lab of CAD&CG at Zhejiang University.

References

[1] W. Li, Y. Lu and W. Lu, "Collaborative Computer-Aided Design – Research and Development

Status", *Proceedings of CAD 04 Conference*, Pattaya, Thailand, May, 2004, pp. 24-29.

[2] F. He and Y. Liu, "Towards Unconstrained Concurrent Control and Naming Consistency in Replicated Collaborative Solid Modeling", *International Conference of CAD&CG'2003*, Macao, China, Nov., 2003, pp. 396-397.

[3] E. VandenBerg "Collaborative Modeling Systems", *Technical Report*, Delft University, 1999.

[4] R. Bidarra, E. VandenBerg and W.F. Bronsvoort, "Web-based collaborative feature modeling", *Proceedings of Solid Modeling '01 - Sixth ACM Symposium on Solid Modeling and Applications*, Ann Arbor, MI, USA, Anderson, D.C. and Lee, K. (Eds.), June 6-8, 2001, pp. 319-320.

[5] X. Zhou, J. Li, F. He, and S. Gao, "Web-based synchronized collaborative solid modeling system.", *Chinese Journal of Computer Integrated Manufacturing Systems CIMS (in Chinese)*, Nov., 2003, 9 (11), pp. 960-965.

[6] U. V. Lukas, " Collaborative Geometric Modeling using CORBA Services", *Advance Proceedings of the ECSCW'97 Workshop on Object Oriented GroupWare Platforms OOGP'97*, Lancaster, UK, 1997, pp. 91-92.

[7] A. Stork and U. Jasnoch, "A Collaborative Engineering Environment", *Proceedings of the TeamCAD97 Workshop on* Collaborative *Design*, Atlanta, USA, 1997, pp. 25-33.

[8] A. Stork, U. V. Lukas and R. Schultz, " Enhancing a commercial 3D CAD system by CSCW functionality for enabling co-operative modeling via WAN", *CD-ROM Proceedings of 1998 ASME Design Engineering Technical Conference*, Sacramento, USA, DETC1998/CIE-5711.

[9] X. Chen and C. M. Hoffmann, "On editability of feature-based design". *Computer-Aided Design*, 1995, 27(12), pp. 905-914.

[10] V. Capoyleas, X. Chen and C. M. Hoffmann, "Generic naming in generative constraint-based design". *Computer-Aided Design*,1996, 28(1) pp. 17-26.

[11] J. Kripac, "A mechanism for persistently naming topological entities in history based parametric solid models", *Computer-Aided Design*, 1997, 29(3), pp. 113-122.

[12] J. Wu, J. Zhang, X. Zhang and J. Zhou, " A face based mechanism for naming, recording and retrieving topological entities", *Computer-Aided Design*, 2001, 33(1), pp. 687-698

[13] D. Marcheix and G. Pierra, " A Survey of the Persistent Naming Problem" , *Proceedings of ACM Solid Modeling and Applications*,2001, pp. 13-22.

[14] R. Bidarra and W. F. Bronsvoort, "Persistent Naming Through Persistent Entities", *Proceedings of Geometric Modeling and Processing*, July, 10-12, 2002, pp. 233- 240

[15] R. Bidarra, J. Paulos, and W. F. Bronsvoort, "A Feature-base Solution to the Persistent Naming Problem", *Computer-Aided Design & Applications*, Vol. 2, Nos. 1-4, 2005, pp 517-526.

[16] Z. Chen, S Gao, F. Zhang and Q. Peng, "An Approach to Naming and Identifying Topological Entities", *Chinese J.Computers* 2001,24(11), pp. 1-8.

[17] D. Mun, S. Han, Kim J. and Y. Oh, " A set of standard modeling commands for the history-based parametric approach", *Computer-Aided Design*, 2003, 35, pp. 1171-1179.

[18] J. Lee, H. Kim, S. Han B. and Park S. B., " Network-centric Feature-based Modeling", *Proceedings of Pacific Graphics 99, IEEE Computer Society*, Los Alamitos. 1999, pp. 280-289.

[19] K. Ramani, A. Agrawal, M. Babu and C. Hoffmann, " Multi-Client Collaborative Shape Design System with Server-based Geometry Kernel", *Journal of Computing and Information Science in Engineering*, June, 2003, 3(2), pp.170-173.

[20] B. Liao, F. He and S. Jing, " Replicated Collaborative Solid Modeling and Naming Problems", *The 9th International CAD/Graphics Conference*, Hong Kong, 2005.

[21] 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 Transactions on Computer-Human Interaction*, 1998, 5(1), pp. 63-108.

[22] L. Lamport, " Time, Clocks, and the Ordering of Events in a Distributed System" , *Communication of ACM*. 1978, 21(7), pp. 558-565.