Journal of Applied Sciences

ISSN 1812-5654

science alert

ANSInet
an open access publisher
http://ansinet.com

Journal of Applied Sciences 13 (22): 4943-4948, 2013
ISSN 1812-5654 / DOI: 10.3923/jas.2013.4943.4948

A Design Command Representation Scheme for Parametric Design in Replicated Instant Collaborative Cad Systems

Shu-xu Jing and Ting-lei Zhang
School of Information Engineering, Chang'an University, Xi'an, China

Abstract: Collaborative CAD aims to break through limitations of geographically distributed orgnizations and designers that use heterogenous legend CAD systems for products development, which can reduce time-to-market for new products. Replicated architecture is currently a hot research topic for constructing a collaborative CAD system due to the light network load, fast response and easy adaptation to heterogenous legend CAD systems. For replicated instant collaborative CAD system, more factors than traditional distributed and parallel computing should be taken into consideration. Instant collaborative parametric design, which is still an open challenge in the research community of collaborative design, must be realized before practical applications can be made. This paper proposes a scheme for parametric design in replicated instant collaborative CAD systems. This scheme concerns on several key factors which are optimistic concurrency control strategy, parametric collaborative design commands representation, parametric collaborative persistent naming, collaboration awareness. The proposed scheme has been implemented in our prototype system and works well.

Key words: Collaborative instant design, parametric design, persistent naming, collaborative awareness

INTRODUCTION

Collaborative design aims to break through the limitations of geographical distribution, organization and designers that use heterogeneous legend CAD systems. Collaborative design will lead to high efficiency and reduce the costs of products development (Jing *et al.*, 2009). Collaboration of design can be implemented at different levels and result in different performances. *Design result reusing* emphasizes on reusing design results across heterogeneous legend CAD systems. This is a type of off-line collaboration. Designs achieved in one commercial CAD system could be reused in other ones. Parametric product exchange belongs to this category (Cai *et al.*, 2009). On-line visualized annotation, where a design can be audited and annotated with visualized marking tools. But annotation may lead to misunderstanding between designers and auditors under some situations. On-line instant collaborative design based on complete homogenous or heterogeneous CAD systems. This kind of collaboration works on design procedure directly and designers can communicate without misunderstanding.

The above three kinds of collaborative design can find their applications in corresponding domain respectively. As for on-line instant collaborative design, parametric design method must be employed before practical applications can be made. To our best knowledge, no literature addresses the problem of parametric design for replicated instant collaborative design system that adopts optimistic concurrency control strategy.

RELATED WORKS

Collaborative system emphasis on free and natural interaction among task participants which is different from traditional distributed system. The philosophy of "What You See Is What I See" was firstly introduced in the community of Computer Supported Collaborative Work (CSCW for short). Recently the philosophy of "What You See Is NOT What I See" based on a Model-View-Controller (MVO) model was addressed for some applications in the community of CSCW (Dewan, 2010). For on-line collaborative CAD, existed literature addressed three major issues as concurrency control strategy, system architecture and data transmission among distributed sites. Collaborative design system experienced pessimistic locking based concurrency control strategy and optimistic concurrency control strategy. Optimistic concurrency control is mostly applied in systems of replicated architecture.

Concurrency control strategies: As simultaneous access to the shared document is involved, concurrency control is the most significant problem in collaborative CAD systems. Two major types of concurrency control

Corresponding Author: Shu-xu Jing, School of Information Engineering, Chang'an University, Xi'an, China

strategies, which are optimistic strategy and pessimistic strategy, are employed in collaborative systems.

Pessimistic strategy: Early collaborative design (or collaborative modeling) systems adopted the floor control strategy (Li *et al*., 2004; Ramani *et al*., 2003; Li *et al*., 2007). This type of strategy is pessimistic and allows only one site modifying the shared document at any given time instant. Free and natural interaction is always blocked due to this rigorous control strategy.

Optimistic strategy: Optimistic concurrency control strategy allows simultaneous modification of shared document and the data consistency maintenance is guaranteed by conflict detection and Operation Transformation (OT for short) algorithms (Jing *et al.,* 2009; Dewan, 2010; Tang et al., 2007; Li *et al*., 2008). Obviously, optimistic strategy provides faster response and more natural interaction performance for collaborative design and conflicts arise very frequently as well.

It is also obvious that conflict avoidance is much easier to handle than conflict detection and solving. Optimistic concurrency control must pay the costs of conflicts solving for the better interaction performance provided.

System architectures: System architecture means data and design function configuration of the whole collaboration system. Two major architectures, which are centralized and replicated, can be found when a collaborative design system is constructed.

Centralized architecture: Client/Server application is typical. A central server is responsible for maintaining the shared document, performing design functions and broadcasting the design result to all clients. Clients just act as visualized interface where design commands are generated and transmitted to central server. The problem of such an architecture is that the server always becomes the"bottle-neck" and can't respond to clients in time.

Replicated architecture: All sites are equally configured with full design functions and each keeps a copy of the shared document. Design commands are generated and executed locally immediately and then pass to all remote sites for remote execution. Better interaction performance can be achieved with the cost of data consistency maintenance. There are also mixed architectures that trade off between the above two where the configuration of design functions and shared document slightly differs.

Data transmission among collaborative sites: Communication among sites is important in collaborative design system. For centralized architecture, Up-to-date geometric data must be transmitted to all clients for synchronization; the network load is heavy even with incremental geometric data transmission techniques employed. Another type of data transmitted over network is design commands in text form, which is employed in replicated architecture. Design command contain rich semantic information and is very compact compared with pure geometric data, the network load is greatly alleviated. Based on above analysis and our previous research (Jing *et al.,* 2009; Cai *et al.,* 2009; Jing *et al*., 2007; Li *et al*., 2009), we propose the framework for parametric design in replicated instant collaborative CAD systems.

PROPOSED FRAMEWORK

Replicated architecture and optimistic concurrency control strategy are employed in our proposed framework. We will elaborate key aspects as Optimistic concurrency control strategy, parametric collaborative design commands' representation and collaborative parametric persistent naming and collaboration awareness.

Optimistic concurrency control strategy: Every site is a homogenous or heterogeneous CAD system and configured with full CAD functions and keeps a copy of shared document. Any site has two states, which are viewing state and modifying state. Under viewing state, the site works as following:

- The site can't do any operation that will modify the copy of shared document. If the site wants to modify the shared document, an explicit local locking operation must be issued and shifts from viewing state to modifying state.
- The site can do any operations that don't modify the copy of shared document. By applying the Model-View-Controller model and "What You See is Not What I See" philosophy(Dewan, 2010), the site can view and control the shared document with any needed methods.
- The site executes design commands received from remote sites and the copy of shared document will be modified as a consequently. Meanwhile, the modification from remote sites can be optionally awared by the site, where optionally means that the site knows what happened to the shared document and then decides whether to notify the designer explicitly or not.

Under modifying state, the site works as following:

- After the local copy of shared document is locked, the site can issue design command locally and executes the command immediately.
- Received design commands from remote sites must be cached while local copy of shared data is locked.
- An explicit local unlocking operation must be issued and shifts from modifying state to viewing state and execute the cached design commands.
- The site transmits the locally issued design command to all remote sites.
- The site receives design commands from remote sites and executes them in transformed form document state.

Parametric collaborative design commands' representation: Two types of design commands, which are feature adding commands and parameter modifying commands, must be taken into consideration when parametric design method is applied. Every site keeps a design command log of all locally and remotely issued commands.

Identification of design commands: Different from stand-alone CAD system, design commands (called DC) from all sites of the whole collaborative design system must be identified. Every design command has a global unique id with the form DCid = (siteid, statevector), where siteid refers to the *id* of the site that issued the design command, statevector (refer (Sun and Ellis, 1998) for detailed definition of statevector) recorded the number of the design commands from all sites that had been executed respectively when the command is issued. Combine the command log and statevector, a document state can be determined. For any two design commands DC1 and DC2, the state vector of DC1 and DC2 is equal only when two or more design commands are rigorously parallel (means that DC1 and DC2 are issued at the same document state) otherwise not equal, then siteid is used to distinguish DC1 and DC2.

Feature adding commands' representation: The term "feature" here means design features, such as a hole, a slot and so on. Feature adding commands are represented in the form (DCid, DCname, Fname, ParamList), where DCid is described above, DCname is the name of the command that expresses design semantic, Fname is the name of corresponding geometric object created in the model under design, ParamList is the parameters that determine the shape and position of the feature added. For example, a hole on a rectangular face needs to refer two edges and two corresponding distance for positioning purposes. Every modifiable parameter has a name and is represented by Pname. If the designer wants to modify a parameter, he can accomplish it through visualized interface, Pname expresses the parameter to be modified.

Parameter modifying commands' representation: Parameter modifying commands are represented in the form (DCid, DCname, DCtid, Pname; Valorg; Valnew), where DCid is the id of the parameter modifying command, DCname stands for the name of a feature modification command, DCtid is id of the target design command to be modified, Pname stands for which parameter is to be modified. Valorg is the original value of the parameter to be modified and Valnew is the updated value of the parameter. Since identification of every command is unique, then target command can be located by DCtid, then target parameter is located by Pname, update the original value with new value. Finally, reconstruct the geometric model with the updated commands log. Parameter modifying command is recorded in the commands log.

Parametric collaborative persistent naming: Parametric collaborative persistent naming originated from traditional persistent naming for topological entities (Jing *et al*., 2007) but is more complex. Figure 1 as an illustration, for the positioning of the hole parameter r and h determines the size of the hole, d1 and d2 together with e1 and e2 determines the position of the hole. All these parameters will be recorded in the design commands for feature adding. How to uniquely assign ids or names persistently to faces, edges and vertices is the classical persistent naming problem in CAD. Persistent naming is difficult due to there exit situations that topological entities are symmetric (for example two vertices of an apple like

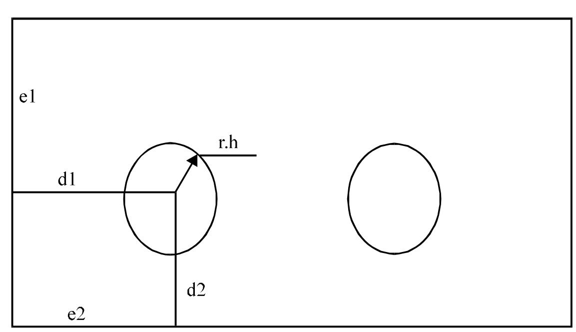

Fig. 1: An illustration of persistent naming

object) and difficult to assign unique names to them, so geometric measurement of the model under design is employed in most cases.

Parametric collaborative persistent naming scheme must support the concurrent execution of design commands as feature adding and parameter modification. A big challenge is that parameter modification will result in change of the geometric measurement of a model. So the dependency on geometry for naming must be eliminated. Taking above considerations into account, three steps to solve this problem is as following.

Collaborative unique naming for topological entities: Based on our previous work (Jing *et al.*, 2009), unique naming for topological entities generated during design procedure can be achieved through involving collaborative information in names of topological entities. Any topological entity Te has an unique persistent name Name(Te), Name(Te) is determined by the document state (DocState for short), where the topological structure and geometric dimensions of the model under design is the key points, when Name(Te) is generated. This fact can be expressed by function (DocState,Te) > Name(Te). Name(Te) is recorded in the design command DC.

Exchange format of topological entities' name: Simplified name format, which only includes StateVector, topological entities' type (face, edge and vertex) and it's geometric information (always academically called Boundary information), is adopted for exchange over network. The format is (statevector, Ele_type, B_rep_info). Statevector and DocState have one to one relationship, expressed by statevector <> DocState. Simplified name of Te is determined by DocState, So the function (DocState, Te) > (statevector, Ele_type, B_rep_info) is satisfied. Why we adopt this simplified name format is for easy extension to heterogeneous legend CAD systems.

Eliminating dependency on absolute geometric information for naming: For any given topological entity Te in a given geometric model, its persistent name Name(Te) and its simplified format (statevector, Ele_type, B_re_ info) has one to one corresponding relationship. We express this relationship with Name(Te) <> (statevector, Ele_type, B_re_ info) and this relationship can be completely computed at any site locally with given document state and topological entity. That is (DocState, Te)⇒ [Name(Te) <> (statevector, Ele_type, B_re_ info)].

Due to concurrent parameter modification, the geometric measurements of shared model will be changed. But if the topological structures are kept unchanged, then corresponding topological entities Te at different sites have the same persistent name Name(Te). Figure 2 as an example, length of the block is modified from 25 units to 50 units, geometric measurement of edge e is changed. If two commands DC are issued upon the model on left and refereed e, it can't be correctly found in the model on right based on simplified name. A temp geometric model equal to the model on left is constructed to corresponding Tes. For an example, two design commands DC1 and DC2 are issued concurrently at site S1 and S2 respectively, which means that DC1 and DC2 are issued at the same DocState. DC1 modified the size of a feature F, DC2 referenced a topological entity Te in F before DC1 is executed just like the situation in Fig. 2. The document under which DC2 referenced Te is expressed by the statevector associated with DC2 (see definition of DCid). When DC2 to be executed at *S*1 at new document state DocState', a temp geometric model will be constructed according to the statevector associated with *DC*2. According to (DocState, Te) > (DC1, Ele_type, B_re_ info), get the temp referenced *Te* through the simplified name format in the temp geometric model. Then generate Name(Te), which is recorded in the command log at S1. Finally, find the topological entity whose name is Name(Te) under DocState,' and that is just the target topological entity should be referenced by DC2.

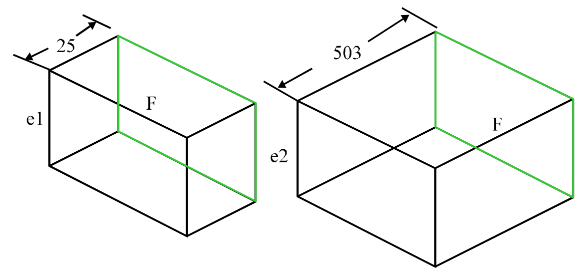

Fig. 2: An illustration of Parameter modification

Another situation is that the topological structure is changed due to concurrent execution of design commands. According to (Rappoport *et al*., 2006), topological entities' corresponding can be achieved through geometrical cover relationship (Union, Intersection, Subtraction relationship) of them. For an example, DC1 and DC2 are concurrently issued commands and DC2 referenced entity Te, which is split into Tea and Teb due to the execution of DC1. According to the statevector associated with DC2, construct the temp geometric model, compute the B_rep_info of Te in temp model and then finding the correspondence relationship with Tea and Teb.

COLLABORATION AWARENESS

Collaborative awareness is significant to help designers communicating to each other in a collaborative design session. Three types of awareness are considered in this scheme, they are design procedure awareness, implicit conflict awareness and version split awareness.

Design procedure awareness: A collaborative session participant maybe interested in a specific designer's work or a specific feature's design procedure, so two types of procedure awareness should be taken into consideration.

Designer oriented design procedure awareness: At a given site S, a collaborative session participant A is interested in designer B's work. S should provide tools for A to view B's design procedure. For this purpose, S find all the design commands DC_{SB} issued by B in the command log. Some commands in DC_{SB} depend on commands issued from other sites DC'_{SB} directly or indirectly. Construct the closure of $\Sigma(DC_{SB} \cup DC'_{SB})^*$, recorded as DC_S, which means that no commands in DC_S depends on other commands outside of DC_S. Finally, reconstruct the design procedure with DC_S^* .

Feature oriented design procedure awareness: At a given site S, a collaborative session participant A is interested in the design procedure of a specific feature H (A hole feature), S should provide tools for A to view the design procedure of H. For this purpose, S Select H by through visualized interface, then find all the design commands DC_{SH} according to Name(H) (is Fname in commands)that contributed to the construction of H in thecommand log. Finally reconstruct the design procedure with DCSH.

Implicit conflict awareness: In a collaborative design session, there exist implicit conflicts when more than one designer issued design commands that has the same design semantic. For an example, two round slots feature is added concurrently and only one is actually needed, so the correct design can't be achieved unless this implicit conflict can be detected and solved. One simple way to solve this problem is by checking the DCname same DCname has the same design semantic. Another more reliable way is checking the result of design, same feature has the same topological structure in the geometric model. when implicit conflict occurred in a collaborative design session, it should be detected and notify all the designers or just designers related for one of them to undo his design.

Multi-version awareness: There exist unsolvable conflicts in collaborative design; designer A and B both insists that their design is better so none of them is willing to undo his work. Under this situation, two versions of the design are kept and notified to all participants. Later, one of them maybe confesses that the other's design is better; he can choose to cancel his own work and merge the split versions of the design. Version split and merge is a challenge problem in collaborative systems and we willnot elaborate this problem in this paper.

CONCLUSION

As for instant collaborative design, many prototype systems that adopt optimistic concurrent control strategy have been proposed recent years, but the parametric design capability which is of very important significance is still absent. Parametric design capability must be at least partially realized before instant collaborative design can find its practical applications. And achieving parametric design capability in instant collaborative design is a big challenge as well. In this paper, we propose an integrative scheme that taken the key aspects as Optimistic concurrency control strategy, Parametric collaborative design commands' representation, collaborative parametric persistent naming and collaboration awareness for parametric design into account. The proposed scheme has been implemented in our prototype system and works well.

ACKNOWLEDGEMENT

This research is supported by the Fund for Basic Scientific Research of Central Colleges, Chang'an University(No.2013G1241108) and Natural Science Foundation of Shaanxi Province(No. 20120849).

REFRENCES

Cai, X. and F. He, X. Li, Z. Huang and X. Chen, 2009. Heterogeneous feature extraction and reconstruction in collaborative design. J. Image Graph., 14: 2615-2618.

Dewan, P., 2010. Increasing the automation of a toolkit without reducing its abstraction and user-interface flexibility. Proceedings of the 2nd ACM SIGCHI Symposium on Engineering Interactive Computing Systems, June 19-23, 2010, Berlin, Germany, pp: 47-56.

Jing, S., F. He and H. Liu, 2007. A survey of persistent naming problem for topological entities. J. Comput. Aided Des. Comput. Graph., 19: 545-552.

Jing, S.X., F.Z. He, S.H. Han, X.T. Cai and H.J. Liu, 2009. A method for topological entity correspondence in a replicated collaborative CAD system. Comput. Industry, 60: 467-475.

Li, M., J.Y.H. Fuh, Y.F. Zhang and S. Gao, 2008. Adaptive granular concurrency control for replicated collaborative feature modeling. Proceedings of the 12th International Conference on Computer Supported Cooperative Work in Design, April 16-18, 2008, Xi'an, China, pp: 116-122.

Li, M., S. Gao and C.C.L. Wang, 2007. Real-time collaborative design with heterogeneous CAD systems based on neutral modeling commands. J. Comput. Inform. Sci. Eng., 7: 113-125.

Li, W.D., S.K. Ong, J.Y.H. Fuh, Y.S. Wong, Y.Q. Liu and A.Y.C. Nee, 2004. Feature-based design in a distributed and collaborative environment. Comput. Aided Des., 36: 775-797.

Li, X., F. He, X. Cai, Y. Chen and H. Liu, 2009. Using procedure recovery approach to exchange feature-based data among heterogeneous CAD systems. Proceedings of the 13th International Conference on Computer Supported Cooperative Work in Design, April 22-24, 2009, Santiago, Chile, pp: 716-721.

Ramani, K., A. Agrawal, M. Babu and C. Hoffmann, 2003. CADDAC: Multi-client collaborative shape design system with server-based geometry kernel. J. Comput. Inform. Sci. Eng., 3: 170-173.

Rappoport, A., S. Spitz and M. Etzion, 2006. Two-dimensional selections for feature-based data exchange. Proceedings of the 4th International Conference on Geometric Modeling and Processing, July 26-28, 2006, Pittsburgh, PA., USA., pp: 325-342.

Sun, C. and C. Ellis, 1998. Operational transformation in real-time group editors: Issues, algorithms and achievements. Proceedings of the ACM Conference on Computer Supported Cooperative Work, November 14-18, 1998, Seattle, WA., USA., pp: 59-68.

Tang, M., S.C. Chou and J.X. Dong, 2007. Conflicts classification and solving for collaborative feature modeling. Adv. Eng. Inform., 21: 211-219.