Collaborative Design Environment with Multiple CAD Systems

J. Y. Chen [1], Y. S. Ma[2], C. L. Wang [3] and C. K. Au[4]

[1]Nanyang Technological University, Chen0077@ntu.edu.sg
[2]Nanyang Technological University, mysma@ntu.edu.sg
[3]Chinese University of Hong Kong, cwang@acae.cuhk.edu.hk
[3]Nanyang Technological University, mckau@ntu.edu,sg

ABSTRACT

The existing collaborative systems are restricted on one specific CAD system and do not support multiple systems. However, employing various CAD systems to facilitate the product design process is not uncommon. This paper presents an approach for collaborative design environment to support a set of heterogeneous CAD systems such that an object can be geometrically co-modelled simultaneously among a team of designers at geographically distributed locations. A command-based solution is proposed to deal with the issues identified in developing such collaborative design system. Operations are used as the transmission data type to realize the collaboration. A prototype is implemented to show the feasibility of the solution.

Keywords: Collaborative design, modelling operation, interoperability

1. INTRODUCTION

The current CAD systems support single user only, however, several engineers are involved in the development of a product in practice. This is obvious for complex products that consist of many parts, but it can even be the case for relatively simple products. Indeed, product development teams nowadays include multiple parties, who are often geographically distributed. They need to work on the same product. Also, collaboration among team members plays an increasingly important role in solving design conflicts in the early design stage. Thus, an environment supports collaborative design with various popular CAD systems is a desideratum.

Collaborative CAD is considered as the next generation CAD system and has received more and more attentions. Most of previous research works [5-7] in this area concentrated on incorporating networking functions into the traditional CAD systems in order to enable the mechanisms of collaborative design. Most systems are restrictive since they are developed based on a specific CAD system.

The objective of this research is to investigate and develop the design function collaboration mechanisms for an Internet based design system. The targeted co-design system is to enable the active participation of various members in a design team either within an organization or between various organizations. The major requirement of this system is to support co-modelling using a set of heterogeneous CAD systems. In addition to visualizing the product data, the design team members are able to perform geometric modelling and modification simultaneously.

2. ISSUES

Three major issues are identified in developing the Internet-based collaborative design system. They are 1) data transmission, 2) synchronization and consistency, and 3) concurrency. These issues have been encountered in the previous research works [5-7] based on a specific CAD system. However, they are more complicated in this project because they should be addressed in an environment that supports different CAD systems.

2.1 Data Transmission

Owing to the large sizes of CAD files, data transmission through Internet is very time-consuming and unreliable. It is intolerable for the designer to wait for a long time for updating large-size CAD models across networks.

To reduce the network-load, an appropriate way to represent CAD data is needed. Ideally, collaboration communication should not only contain CAD information and convey design intent, but also have the size of data packets as small as possible. Incremental transfer is an alternative way to reduce the network-load. Only the modifications, instead of the whole CAD model, are transferred incrementally during the design process. The repository of the entire product can be centralized with a server as suggested in [10]. Then the network-load is greatly reduced.

2.2 Synchronization and Consistency

Synchronization involves the handling of design modifications by multiple users. These changes will eventually have to be sent to and checked by other users to maintain the consistency of CAD data [8].

In this research, it has been assumed that there are different CAD systems employed to carry out the modelling functions in collaboration. The formats of geometric model created by various CAD systems are different. Thus two aspects need to be considered in order to address the issue of synchronization and consistency. Firstly, differently formatted geometric models are incomparable, and then the problem is how to determine whether these models are synchronized and consistent. Secondly, when some modifications are made on one geometric model, how the corresponding modifications to be propagated to other differently formatted geometric models in order to maintain the consistency of each local product model.

2.3 Concurrency

In the field of collaborative CAD systems, concurrency problems are encountered when different users try to manipulate one or more objects in the design simultaneously [8]. Concurrency control is a crucial issue in development of collaborative CAD system. If this issue is not addressed properly, serious problems such as inconsistency of geometric model and deadlock (which means the processes infinitely wait for each other) can arise.

An effective mechanism is needed to detect and solve the concurrency conflicts among various operations performed on the model from various parties during the design process.

3. PROPOSED SOLUTION

In a collaborative product development environment, individual companies or teams involved in a given product development activity should be able to collaborate using a set of heterogeneous systems. They need to exchange information meaningfully and pass knowledge between various phases in the product development cycle. Hence the research scope is extended from a specific CAD system to multiple CAD systems for co-modelling. This is hard to realize due to the lack of interoperability among various CAD systems. The current state of art for data exchange has the technologies to support a few international standards, e.g. STEP and SAT, based on solid models. However, higher level information, including features, design intent, constraints, associativity and semantic patterns, is lost in the data exchanging process. Dynamic collaboration among different systems is still a technical challenge which requires more attentions. Unified feature modelling approach proposed in [2] offers a framework to enable feature level interactions in a heterogeneous environment. However, to develop the targeted system, the issue of interoperability to enable dynamic high-level CAD functional collaboration and information sharing/exchange among different systems, has to be addressed.

In this paper, a command-based solution is proposed to deal with these issues. The structure of the integration system is shown in Fig. 1. For every client CAD software, a manager application is to be developed to encode and decode the collaborative operations of a product model, where the commands are encoded by XML [3] format to be transferred. Compared with other collaborative design solutions, e.g. CMS [1], webspIFF [8], ideally, the implementation of the proposed solution involves only integration works without going deep into the kernel source of the selected CAD systems, and the powerful functions of each local system can still greatly benefit client users.

The major idea of this integration solution is to utilize the current resources as much as possible. As shown in Fig. 1. – the structure, some manager applications are developed to attach on the selected commercial CAD systems; then the manager applications take the duty of data encoding, data transferring, and data decoding – the collaboration between users and systems. Obviously, the related product and process models have to be stored into a database. A feature oriented product database has been reported in [10], which is still an on-going parallel effort conducted in NTU.

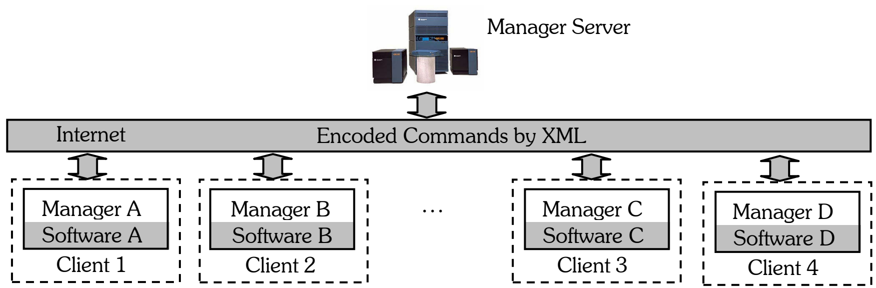


Fig. 1. Structure of the integrated collaborative design environment

The proposed solution uses "operation" as the transmission data type to realize CAD data exchange/sharing among different CAD systems. As illustrated in Fig. 2., during the process of collaborative design, once one designer with CAD system S_A makes some modifications, each corresponding operation, namely the local operation P_A, is captured and mapped into a standard operation by the manager M_A. And the standard operation is encoded by XML format and transferred to the server. On the site of another CAD system, S_B, when the command reaches, the manager M_B first decodes it to interpret the standard operation, then maps it into local operation P_B which is an executable operation for CAD system S_B. Through executing local operation P_B by CAD system S_B, the same modifications can be achieved on CAD file F_B. By this way, the CAD file F_B is updated.

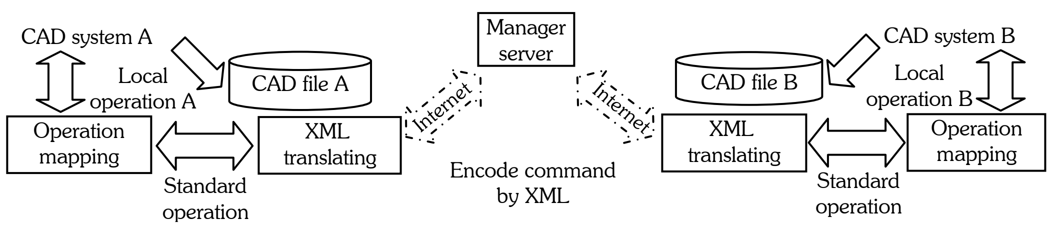


Fig. 2. Schematic figure of using "operation" to exchange information among different CAD solutions

With this strategy, no special CAD system is needed for the collaborative design. Only an operation manager application attached to each CAD system is to be developed on the involved CAD systems.

4. DYNAMIC CAD DATA SHARING/EXCHANGE BY OPERATIONS

4.1 Operations in CAD Software

In this work, an operation in a CAD system is defined as a set of related commands which are provided by software to achieve functional manipulation on entities. It is directly used to support the interface of a CAD system. It carries the design data and its intent. For example, part design with a CAD system is an interactive process between the designer and the CAD system. Operations provided by a CAD system act as interfacing functions for the designer to access the product entities and their related functions. The designer uses the operations to design the part step by step. During the design process, the designer converts the design concepts and rules into a group of operations. Through these operations, the design data and design intent are created and maintained by the CAD system.

Usually, a CAD system may have many types of operations. From the viewpoint of operational functions, the operation can be classified into two types, namely geometry-related and non-geometry related operation, as illustrated in Fig. 3. A geometry-related operation is the operation that will affect the geometric model and it will either create new objects into the geometric model or modify the existing objects. Non-geometry related operation could be divided into "auxiliary" operation and "additional" operation. "Auxiliary" operation is mainly to facilitate the designer to create geometric model and will not affect the geometric model, such as layer management and view manipulation. The left operations can be classified into "additional" operation, such as file management.

The geometry-related operations can be classified into feature-related and low-level operations according to the objects that they create or modify. The low-level operations are to create or to modify low-level entities of a CAD system, such as points, lines and faces. The feature-related operations are to create or to modify the features in a CAD system.

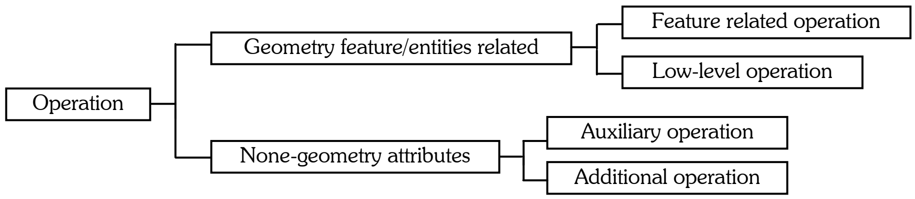


Fig. 3. Classification of operation in CAD system

A more important characteristic of an operation is its execution. It is obvious that the operation must be executed among different CAD systems via the different functions provided. It means that operations initiated by one CAD system, can be carried out with some equivalent functions in another. By mapping appropriate operation methods catered for different CAD systems, operations are interoperable with equivalent definitions and functions across multiple CAD systems directly via a standard scheme. By executing the related operations, the same product model can be generated. For example, by executing the view manipulation operation, the same screen display can be achieved.

4.2 Definition of A Base Operation

A feature can be viewed as a pre-defined data structure that provides an easy way for a designer to manipulate the geometric model, while operations are the dynamic processes to manipulate features. As shown in Fig. 4., operation in collaborative design layer is the basic element for collaboration which is on top of feature representation layer, with effective functions and sequence encapsulated. Multiple CAD systems can be integrated through functional mapping and high-level data exchange.

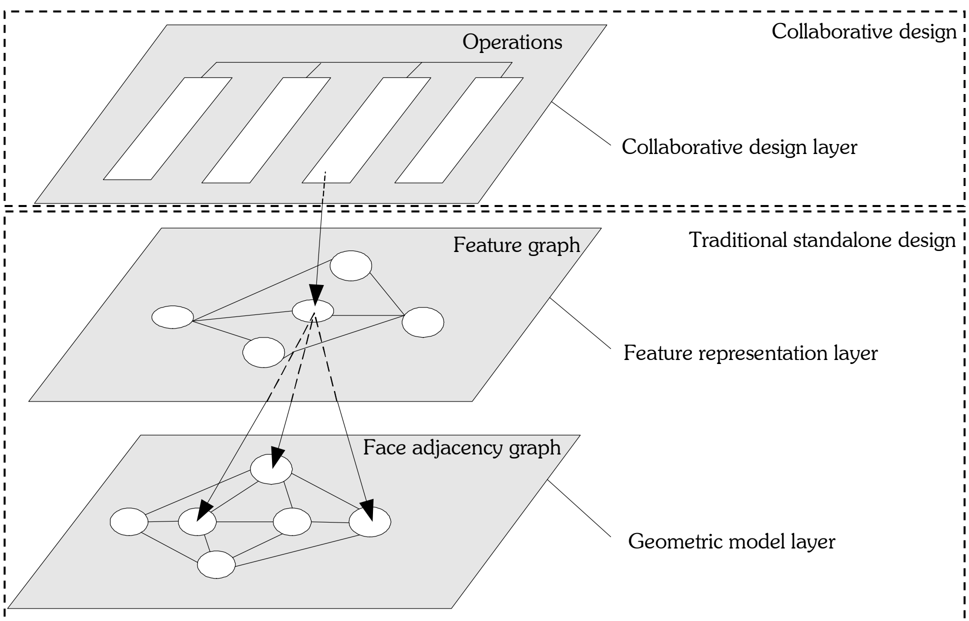


Fig. 4. Element of collaborative design

The operation type has three types of constituents, namely 1) functions, 2) design sequence and 3) target and reference entity ID. The function of an operation encapsulates enough CAD information to result in the changes of product model. Since traditional standalone design usually employs features as the design elements, as shown in Fig. 4., for a geometry-related operation, its function is mainly to create a feature or to modify an existing feature. Thus the CAD information encapsulated in function is usually high-level semantic messages to instantiate a feature or to modify a feature. The sequence of an operation is generated naturally during the design process which makes the operation a natural unit. The sequence of the operations equips the operation type with strong time characteristic. Compared to features, operation is more suitable to be the transmission data type for a collaborative application. In contrast to the traditional feature definitions [9], in this work, a feature is defined as an associative constituent of a product model [2] across different domains of engineering; in design aspect, it provides an easy way for a designer to manipulate the geometric model. Feature does not possess time characteristic and is not sufficient to meet the requirement of high interactivity of collaborative design.
The base operation is defined as a generic data type, and is represented as follows:

Base operation class

Operation type: type
Operation attributes: name,
time stamp,
sequence
Operation ID: int
Target object: taget_object_list
Reference object: ref_object_list
Operation parameter: parameters

Check_validity()
Run_Function ()
Query_information ()

Fig. 5. The class of a generic operation object

Fig. 5. illustrates the base class of operation. The type of operation is referred to the classification of operations as shown in Fig. 3. The parameters of an operation will vary depending on the type of operation. For example, if the operation is feature-related one, the parameters will be the information related to a feature, such as feature parameters, location, orientation and constraint. The target objects are those affected by the operation. The reference objects provide the required information in the operation. The method Check_validity is to check the execution of an operation. The individual concrete operation can be derived from this base operation class. Fig. 6. shows the hierarchical structure of operation sub-classes. Here the sketch operation is used to encapsulate the low-level sketch related operation, such as points and lines, to improve the efficiency and effectiveness of collaborative design.

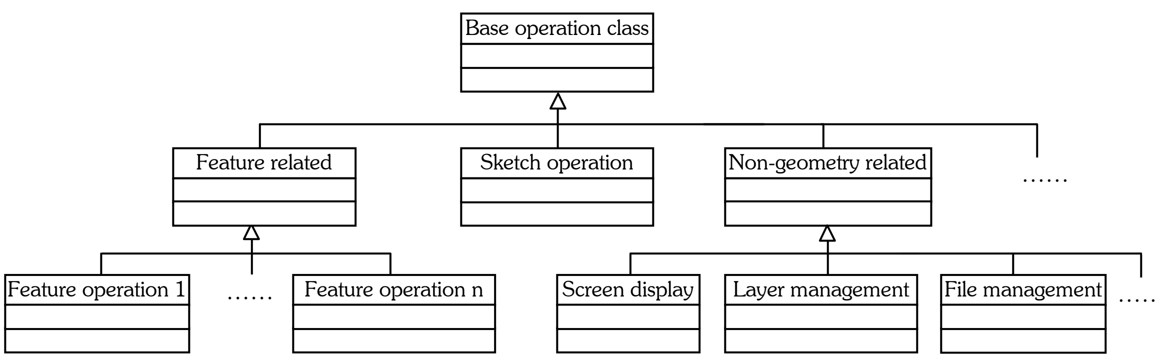


Fig. 6. Operation class hierarchy

4.3 Requirement of Standard Operation

Since different CAD systems may have different ways to achieve an operation, an operation conversion mechanism is needed to address CAD functional interoperability. As illustrated in Fig. 2., a standard operation library in the proposed solution is used to construct a channel for CAD data exchange/sharing among various CAD systems.

The requirements for a standard operation are as follows:

- Not tied to a single vendor software solution;
- Simple and generic;
- Independent to any one product development process;
- Capable of implementing common CAD modelling functions; and
- Extensible to create broader engineering applications.

The standard operations can be employed as the neutral validation criterion. By comparing the standard operations that recorded on each collaborative site, the synchronization of the geometric model can be determined. By maintaining the consistency of the standard operations, the consistency of differently formatted geometric models is achieved. This capability is based on the presumption that all the features of the CAD systems involved can be mapped into the equivalent ones via a feature-level standardization scheme.

5. CASE STUDY

A prototype collaborative CAD system is developed to verify the feasibility of using operations as the transmission data type to realize dynamic CAD data exchange/sharing among different CAD systems. The commercial CAD systems Unigraphics and SolidWorks are selected to carry out modelling routines on client side. The prototype system employs features as the design elements, thus only the operations which create or modify features, are transferred. At present, only a simple and strict ‘control baton’ mechanism is employed to control and schedule the collaborative activities.

5.1 Mapping to Standard Operations

To address CAD interoperability, a set of standard operations is pre-defined as illustrated in Fig. 7. based on the design form features defined in STEP [4]. These operations are all geometry-related operations. Besides the standard operations, a set of mapping rules is defined for two commercial CAD systems, Unigraphics and SolidWorks respectively. The local operations of these selected CAD systems are mapped to standard operations.

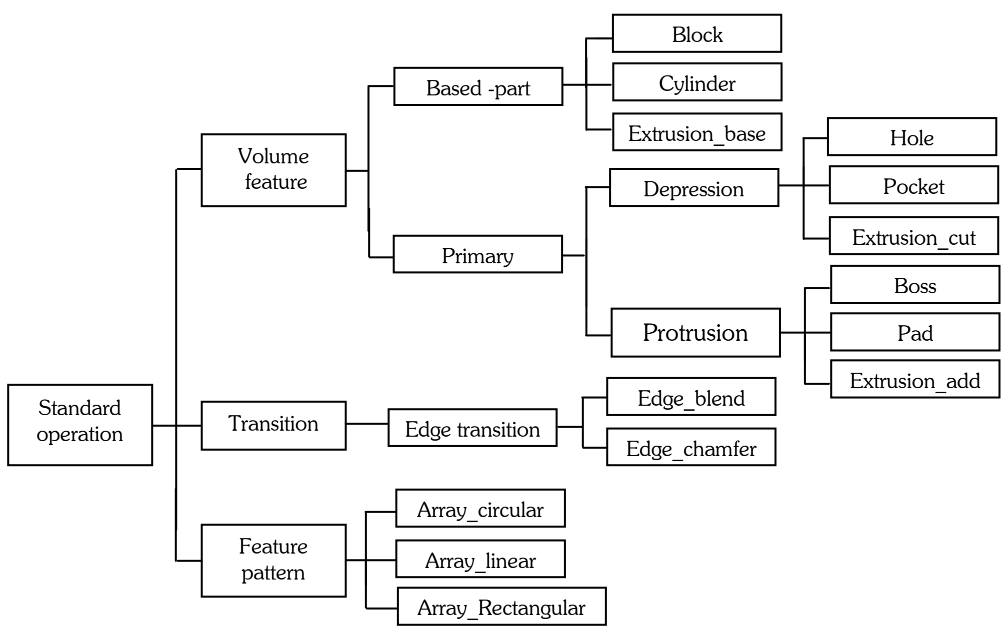
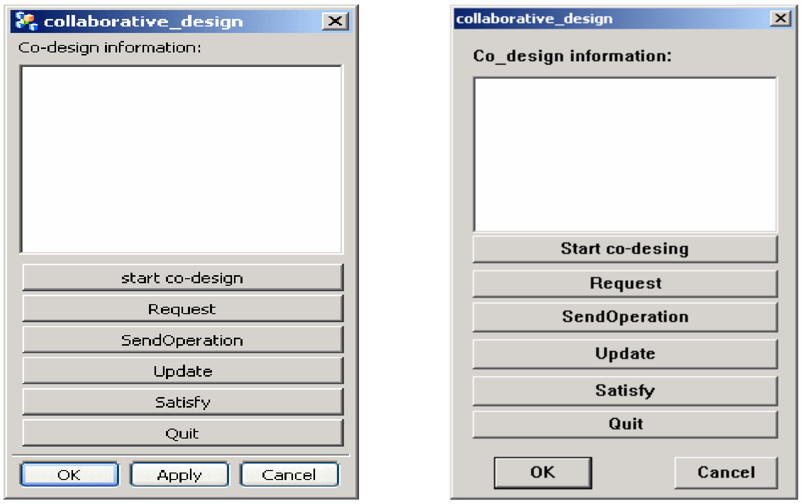


Fig. 7. Some standard operations

5.2 User Interfaces
To enable collaborative design, the user interface is developed on the selected CAD system based on its API. The main dialogue boxes of the user interfaces for Unigraphics and SolidWorks are shown in Fig. 8(a). and Fig. 8(b). respectively.



(a) (b)

Fig. 8. User interface: (a) for Unigraphics and (b) for SolidWorks

5.3 Control Mechanism for Collaborative Design Processes
In the prototype system, a 'control baton' mechanism is employed to coordinate the collaborative activities. Only the user with the control baton can edit the part, the other users are observers. They receive the information from the server to update their parts.

During the process of collaborative design, if one user wants to modify product data, he clicks "Request" button to send request to the server. The server processes the incoming requests and assigns the control baton to the users according to the principle of "first come, first serve". When the server passes the control baton to a user, it sends a message to inform other users to wait. The user with the control baton is the controller and he has the rights to change the product data. After modification, by clicking "SendOperation" button, the corresponding operation will be captured and mapped into a standard operation at the same time. The standard operation will be sent to the server with the control baton and the user becomes an observer. The server will send a message to the previous controller to confirm the operation has been received and deliver the operation to other users. When the user receives the operation from the server, by clicking "Update" button, the standard operation will be mapped into the corresponding local operation and the local operation will be applied on the model on his machine. After the update is finished, every local machine sends a confirmation to the server. If the server does not receive the confirmation, the command will be sent to the machine for one or two more times. If there is still no confirmation from the machine, the machine is assumed to have left collaboration space. The user on that machine needs to re-join the collaboration space. After receiving the update confirmation from all these machines, the server allows all users to start a new round of collaborative design. For any time, when there is a new user joining the collaborative design, the standard operations on the server are utilized to initialize the product model for the new user. If the user is satisfied with the design, he clicks the "satisfy" button to confirm the acceptance. The server will end a co-design cycle only after receiving consent from all the participants. As a result, the same geometric model is generated on each client machine although it is stored in different formats.

5.4 Demonstration
This section demonstrates a prototype system supporting collaborative design with Unigraphics and SolidWorks through a simple scenario. In this scenario, three designers are assumed to build a part collaboratively and a manager server is running separately to coordinate the collaborative activities. The part and the operations that may be used to model the part are shown in Fig. 9. The corresponding standard operations are also shown.

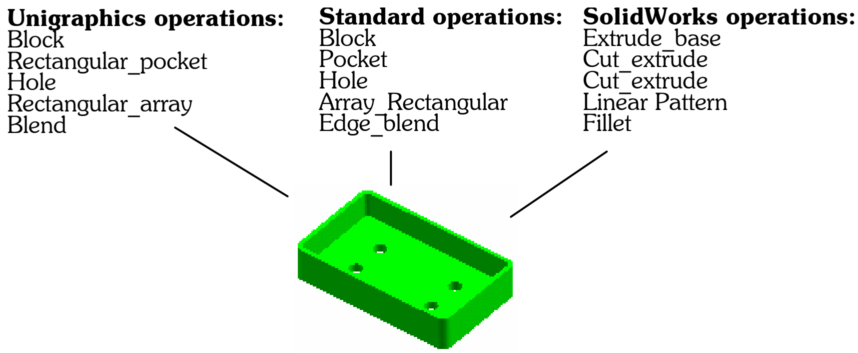
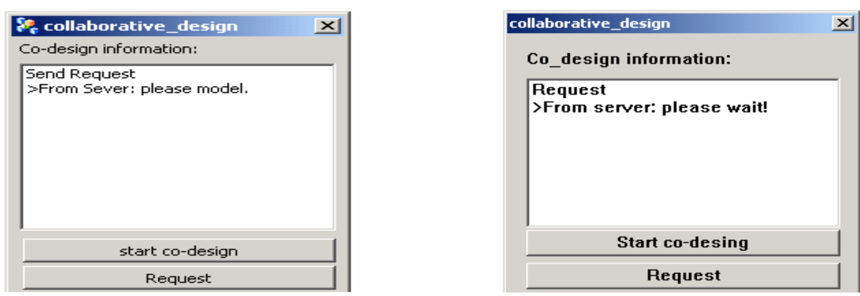


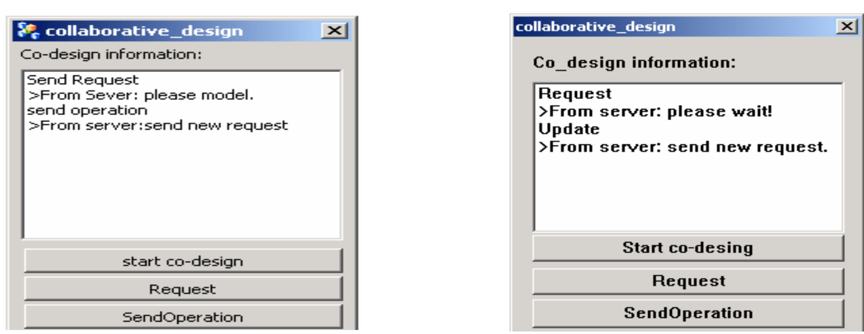
Fig. 9. Part and its operations

Assume that at the beginning, there are two users designing the part collaboratively, named site 0 with Ungraphics, and site 1 with SolidWorks. Site 0 clicks "start co-design" button to send a message to the server to set up collaborative design space and site 1 clicks the same button to join in. At the first round, the site 0 gets the control baton and the server informs site 1 to wait, as shown in Fig. 10. Site 0 uses the "Block" operation to create the base feature in Unigraphics. After finishing the operation, he clicks the "SendOperation" button, and the Block operation of Unigraphics is mapped into the standard operation "Block". The standard operation with the control baton is sent to the server. The server delivers the standard operation to site 1 and sends a confirmation to site 0. Site 1 clicks "Update" button and the standard operation is mapped into local operation "extrude_base" of SolidWorks and executed on the machine of site 1. And update confirmation is then sent to the server. After receiving the update confirmation, the server informs all sites to start a new round of collaborative design (see Fig. 11.).



(a) (b)

Fig. 10. The screen layout of the co-design interface at (a) site 0, and (b) site 1

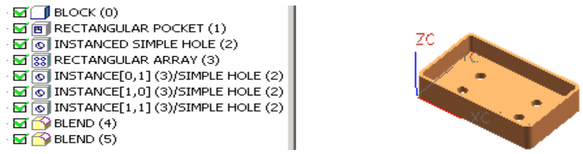


(a) (b)

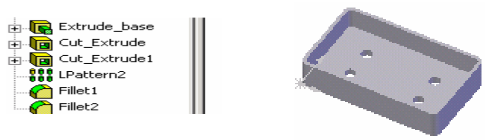
Fig. 11. The screen layout of the co-design interface at (a) site 0, and (b) site 1

At certain time, the third user at site 2 using Unigraphics joins in the collaborative design. He clicks “start co-design” button to send a request to the server. The server will wait until the controller sends the latest standard operation to it. After received the latest standard operation, the server will send the previous standard operations and the latest standard operation to site 2. And the site 2 can request the control baton at next round of collaborative design.

By repeating the process described as above, a part is co-modelled. After all users are satisfied with the part, the server will end the collaborative design session. And through collaborative design, on each user’s machine, the same part, as shown in Fig. 12., will be generated although it stored in different formats. The parts on different machines not only have same geometric data, but also have the equivalent parametric information.



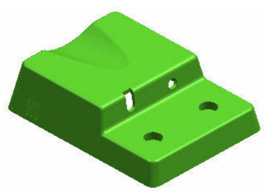
(a)



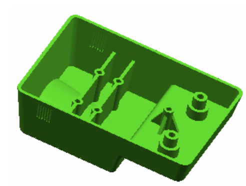
(b)

Fig. 12. The screen layout of result at (a) Unigraphics, and (b) SolidWorks.

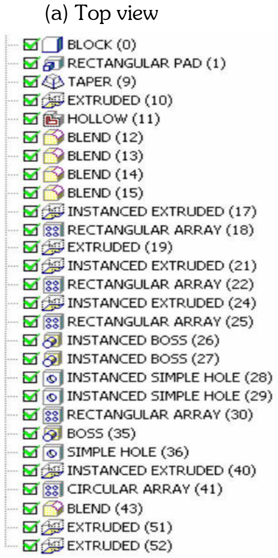
This co-modelling approach is equally applicable for complex products, it can even be the case for relatively simple products. Based on the same prototype system discussed above, some more complex products, such as the one shown in Fig. 13., have been co-modelled successfully through the prototype system.



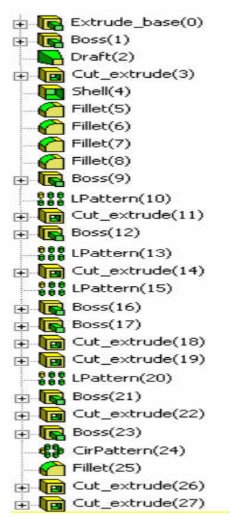
(a) Top view



(b) Bottom view



(c) Feature tree in UngraphicS



(d) Feature tree in SolidWorks

Fig. 13. A part tested on the prototype system

6. CONCLUSIONS

Operation is proposed as the basic element for collaboration information transmission, on top of feature, with effectively operational functions and sequences encapsulated. Theoretically, with the operation standardized, multiple CAD systems can be integrated through functional mapping, and high-level data exchange/sharing can be achieved. Using operations as the major transmission CAD data type possesses the following advantages:

- *Small size for the transferred file.* Nowadays, most CAD software supports feature modelling and parametrical design; features are usually employed as the design elements. During the process of collaborative design, with this proposed method, only high-level semantic messages (such as specifying feature-related operations and data) are transferred between clients and the server. Thus using operation representation can achieve a rather small size of the data file transferred over the network than using other formats, such as original CAD files or STEP files. Hence, this approach can enhance the efficiency and effectiveness of communication in a collaborative CAD system.
- *Complete information.* An operation not only contains the information to create geometric model, but also carries the design intent. Using operations as the transmission data type can realize high-level CAD data exchange/sharing. Furthermore, by transferring non-geometric operations, other kinds of data, such as layer information and screen view can be synchronized among remote users.
- *Incremental transfer.* In a collaborative environment, when one user makes some modifications, the same modifications can be achieved on the machines of remote users only by propagating the corresponding operations to the remote users.
- *Multiple CAD software supportable.* Operation is a special way to represent CAD data and it rarely deals with low-level geometric data. As discussed above, by applying appropriate modifications on the operation, it can be executed by different CAD systems. Compared to complicated geometric data conversion for exchanging, operation level interoperability is more feasible.

This paper has explored the key working mechanisms within a preliminary prototype system. The experimental results from the current simple prototype system have shown the feasibility of the proposed method that uses “operations” as the basic transmission data type to address the functional interoperability. Dynamic high-level CAD information sharing among different CAD systems is realized.

7. REFERENCES

[1] Chan Stephen, Ng Vincent, Collaborative solid modelling on the WWW, *Proceedings of the ACM symposium on Applied Computing*, San Antonio, CA, 1999, pp598-602.

[2] Chen, G., Ma, Y. S., Thimm, G. and Tang, S. H., “Unified feature modelling scheme for the integration of CAD and CAx”, *Computer-Aided Design & Applications*, Vol. 1, Nos. 1-4, 2004, *CAD’04*, pp 595-602.

[3] E. R. Harold, *XML Bible*, IDG Books Worldwide INC., New York, NY, 1999.

[4] Industrial Automation system and Integration — Product Data Representation and Exchange — Part 224: Mechanical product definition for process planning using machining features, *ISO Document ISO TC 184/SC4/WG3 N998*, 2000.

[5] Li W.D., Ong S.K., Fuh J. Y. H., Wong Y.S., Lu Y.Q., Nee A.Y.C., Feature-based design in a distributed and collaborative environment, *Computer-Aided Design*, Vol. 36, No. 9, 2004, pp775-797.

[6] Ouyang Ying-xiu, Tang Min, Lin Jun-cheng, Distributed collaborative CAD system based on Web Service, *Journal of Zhejiang University Science*, Vol. 5, No. 5, 2004, pp 579-586.

[7] Qiang L., Zhang Y. F., Nee A. Y. C, A distributed and collaborative concurrent product design system through the WWW/Internet, *International Journal of Advanced Manufacturing Technology,* Vol. 17, No. 5, 2001, pp315-322.

[8] Rafael, B., Eelco Van den Berg, Willem F. Bronsvoort, Collaborative modelling with features, Proceeding of ASME Design Engineering Technical Conferences, Pittsburgh, Pennsylvania, 200, pp1-11.

[9] Shah J.J., Rogers M., Sreevalsan P. and Mathew A., Functional requirements for feature based modelling systems, *CAM-I Report R-89-GM-03*, CAM-I, Texas, USA, 1989.

[10] Tang, S. H., Ma, Y. S. and Chen, G., “A feature-oriented database framework for web-based CAx applications”, *Computer-Aided Design & Applications*, Vol. 1, Nos. 1-4, 2004, *CAD’04*, pp 117-125.