Journal of Zhejiang University SCIENCE
ISSN 1009-3095
http://www.zju.edu.cn/jzus
E-mail: jzus@zju.edu.cn

Distributed collaborative CAD system based on Web Service*

OUYANG Ying-xiu (欧阳应秀)†, TANG Min (唐 敏),
LIN Jun-cheng (林军呈), DONG Jin-xiang (董金祥)
(*State Key Laboratory of CAD&CG, AI Institute, Zhejiang University, Hangzhou 310027, China*)
†E-mail: oyyx_zju@sohu.com

Received Mar. 26, 2003; revision accepted July 3, 2003

Abstract: This paper presents a novel Web Service based distributed collaborative CAD system employing feature as its collaborative design element and uses XML to define feature operations and communication protocol between the server and the client. To reduce network load and increase response ability of the system, the feature information is updated incrementally on the client. The system supports collaborative designing on heterogeneous platforms. Its framework and communication protocols are analyzed in detail. The experimental results from the developed prototype system showed that it can effectively support collaborative design under the distributed environment.

Key words: Web Service, XML, Collaborative design, Feature
Document code: A **CLC number:** TP391.72

INTRODUCTION

Computer Supported Collaborative Work (CSCW) refers to technologies that use computer technologies, multimedia technologies and network communication technologies to support geographically distributed users to collaborate with design tasks under shared environments. It is argued that the product development period may be shortened greatly when users independently and collaboratively join in the designing, planning and assembly process with the same product under a distributed environment (Wang *et al.*, 2001).

As a new standard for data exchange, XML (eXtended Markup Language) is becoming appealing to worldwide researchers and developers (Rezayat, 2000). It provides an effective means to exchange CAD data. Primarily, as XML is a self-descriptive language and its structure is separated from display, it is easy to validate an XML document and reuse its data with its schema. Moreover, CAD data represented by XML may be shared among different CAD systems. Hence, XML has the obvious advantage of representing; managing, sharing and reusing complicated product data. Jian and Tan (2001) used data hierarchical tree to construct XML DTD for general XML representation by analyzing Express and XML and established correspondence between them. During former work, the authors successfully converted DWG documents to XML documents, and effectively implemented data query and semantic search operations on the converted files.

Web Service, which can be accessed by standardized XML messaging methods via network, is an interface that describes some operations. It is a conception or technique that turns the Inter-

* Project supported by the National Research Foundation for the Doctoral Program of Higher Education of China (No. 2000-033554) and the Natural Science Foundation of Zhejiang Province of China (No. 6001107)

net/Intranet into a virtual computing environment. Heterogeneous clients can access Web Service on the server via standard network protocols such as HTTP and SMTP. The conceptual architecture of Web Services mainly includes three parts: service description, query/discovery of service, and binding/using service. Service description is the description of service interface and of implementation details with WSDL (Web Service Description Language) including data types and operations of Web Service, binding information and its location. Query/discovery of service is the discovery of service, and binding/accessing service occurs when the service requestor binds and accesses service by service description.

The data exchange procedure between the client and the server when the client invokes the service on the server is shown in Fig.1. The request message is sent to the server via the lower network protocol (i.e. HTTP) after the client encapsulates it with XML using SOAP (Simple Object Access protocol). On receiving the SOAP request, the server parses the request, executes the corresponding service and returns the result to the client with SOAP. As Web Service is built under open standard and accepts heterogeneous clients accessing its service, the Web Service based applications can be loosely coupled, com-oriented, and implemented with multi-techniques.

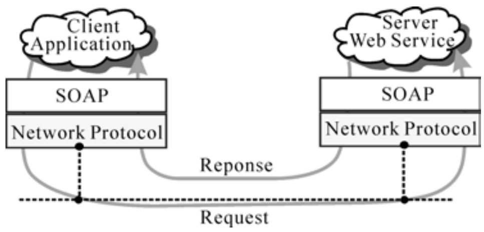


Fig.1 Data exchange between the client and the server when Web Service is invoked

After analysis of current collaborative CAD systems, a novel Web Service based distributed collaborative CAD system is proposed in this paper. The system uses XML to define feature operations and the communication protocol between the server and the client. The system uses HTTP to transfer SOAP objects, enabling it to support multiplatform collaborative design.

REVIEW OF PREVIOUS WORK

Several collaborative modeling prototype systems described in literature will be briefly discussed here. CollIDE (Nam and Wright, 1998), a plug-in for the Allias modeling system, provides some collaborative functionalities. However, the structure of CollIDE poses severe restrictions to crucial collaborative modeling issues. In particular, no special measures were taken to reduce the amount of data sent between the client and the server, resulting in delayed synchronization of operations and of the users' displays. ARCADE (Stock and Jasonch, 1997) reduces data sent between users by adopting the fat-client mode of client-server architecture, and only the change of product model is sent by short textual messages. A drawback of this approach, however, is that it becomes rather complex in the user application, thereby requiring much computational power of the client. CSM (Chan *et al.*, 1999) is also a web-based collaborative modeling system. Within its client-server architecture, the server contains a global model, while every client owns a local copy of this model. When a user has locally modified the model, it is propagated to all of the other users through the server. However, concurrence is managed by a strict token passing manner, which turns the clients into several independent modeling systems just using the same product model alternatively. WebSPIFF (Rafael *et al.*, 2001; Willem *et al.*, 2001) is claimed to be a web-based collaborative modeling system. It adopts the so-called thin client mode of client-server architecture. In WebSPIFF, the server mainly does operations related to solid modeling, and the client mainly deals with user interface operations and result display. Connection between the client and the server is done by direct Socket. When updating the client user interfaces, images used to update client user interface are stored in the SPIFF server, and the server notifies the client(s) to download them through the hypertext transfer protocol. However, this approach increases the data exchange times on the network, and the time delay to update the client user interface. When user frequently operates on the product mode, the client user interface could not be updated in time. CADDAC (Agrawal *et al.*, 2002) employs a three-tier architecture that partitions the collaborative design system into a server-side, a client-side, and a database. In CADDAC, the server controls a central model and does most modeling operations. The client is thin in that it needs to have only a connection to the server and graphics libraries for display. Also, the system is capable of doing collaborative constraints management.

ARCHITECTURE OF DISTRIBUTED COLLABORATIVE CAD SYSTEM BASED ON WEB SERVICE

A novel Web Service based distributed collaborative CAD system using XML to exchange data between the client and server is proposed in this paper. The system's architecture is shown in Fig.2. The client-server type of architecture is adopted, in which interactions between the server and the client are realized by invoking Web Services published by them. The server executes all of the solid modeling operations, such as feature creation and conversion, maintenance of feature validity, and handles concurrency and synchronization of clients. These operations are accessible to the clients in the form of Web Services. The client mainly provides facilities for the user interface display, and converts the user's modeling operations into the invocation of Web Service provided by the server. In order to lower network load, only high-level semantic messages (such as specifying operations and data needed to update the client user interface) are transferred between the server and the client. The data transferred from the client to the server is in the form of CSCW request, which only specifies the request type and corresponding parameters, and the data transferred from the server to clients is in the form of CSCW response containing operation type and B-rep data needed to update client display if necessary. Also, CSCW events, generated during the design process, are transferred between the clients and the server. The client user interface is updated incrementally by XML, i.e., only the changed, not the whole product model data, is transferred. This ensures great reduction of the data amount transferred between the server and the client, decrease in time delay of operation, and enhancement of the system response ability.

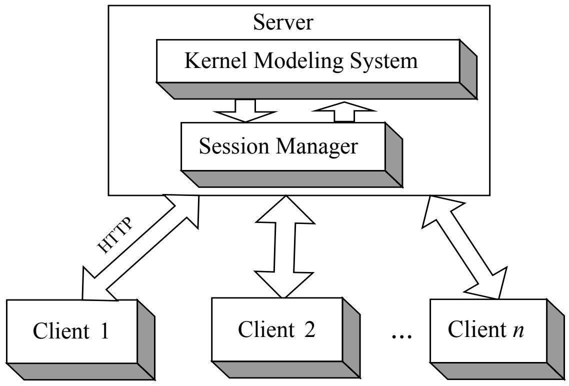


Fig.2 Schematic figure of Web Service based collaborative CAD system

Server

As shown in Fig.2, two main components can be identified on the server: the kernel modeling system and the session manager. The kernel modeling system provides all of the modeling and feature visualization functionalities. A central product model is maintained by the server. An important advantage is that only one central product model is kept in the whole system, avoiding inconsistence between multiple versions.

Session Manager

The server not only executes the operations committed by clients but also coordinates and synchronizes clients' operations in order to make sure that the modeling operations will not conflict. It is the task of the session manager to synchronize session participants to maintain the integrity and validity of solid modeling operations. It provides functionalities to start, join, leave and close a modeling session and manages all information streams between the clients and the kernel modeling system.

As shown in Fig.3, the session manager consists of Request Arbitrator, Client Agent, Client Profile and Session Profile. Each of the session participants is assigned a Client Agent by the session

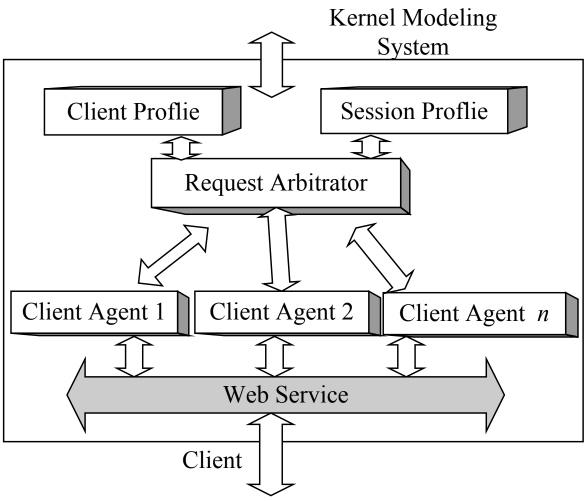


Fig.3 Architecture of the Session Manager

manager. The Client Agent handles all the information between the client and the server, collects and parses client's request and hands it to the Request Arbitrator. The Request Arbitrator then selects one operation and passes the request to the kernel modeling system to execute according to certain arbitrating rules by resorting to the client profiles and the type of the operation. The Request Arbitrator also coordinates concurrency and synchronization of client's operations. Synchronization is the process of propagating evolving data in order to keep the information consistent in a distributed environment. Two types of information can be distinguished here: (1) updated feature mo- del data, and (2) updated state information. After executing the model operation successfully, the server updates the client user display with the modified data incrementally. Updating state information is achieved by triggering a CSCW event, i.e., when the kernel modeling system is idle, an idle event indicating this is triggered by the session manager on the server, and all of the clients are notified. The client profile, created upon the client starting or joining the design session, contains basic information of session participants. Session profile is created when a session is started.

Client

In order to reduce network load and amount of data exchange between the client and the server, user operation such as feature pick-up should be done locally as much as possible on the client side, and only high-level semantic messages, such as specifying operations and data needed to update the client user interface, are transferred between the server and the client. As shown in Fig.4, three components can be identified on the client side: Client Agent, User Interactive Module and Display Module. Interaction between the client and the server is accomplished by the Client Agent, who encapsulates user's operation and information with XML and invokes the corresponding Web Service on the server. It collects and parses XML data coming from the server and passes the parsed data to the display module. It is the task of the User Interaction Module to accept user operations on the client side and deliver to the Client Agent the operations that should be executed by the server.

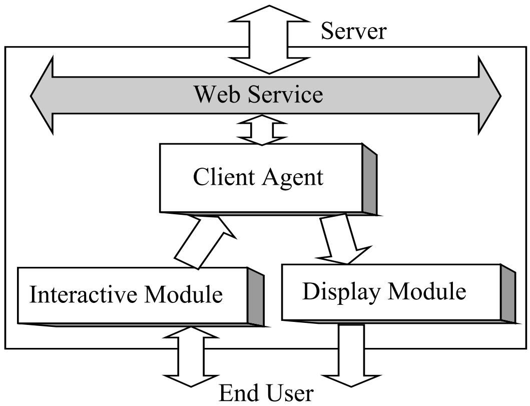


Fig.4 Architecture of the Client

PROTOCOL DESCRIPTION OF THE SYSTEM

We have defined the interaction protocol between the client and the server used in the system. All the information exchanged between the client and the server is defined with XML.

Header

Each operation information in the system proposed in this paper contains a header element. The header is defined as a quintuple: <Version, Time, SessionID, Source, Destination>. Version is the version of the current system; Time is the time tag when the message is sent; SessionID is the ID of the current design session; Source and Destination are the message sender and receiver, respectively.

CSCW Event

CSCW Events generated during the design session, are reported to the design participants of the CSCW events. These events include user starting, joining and leaving session, idle state of kernel modeling system and user adding, editing or removing feature or feature constraints. Besides a header, a CSCW Event contains an EventData field. An EventData is defined in two-tuples: <EventType, EventContent>, where EventType is the type of the event generated and EventContent contains the event in detail.

CSCW Request

During the collaborative design session, collaboration among clients is achieved typically by invoking Web Services on the server. Operation requests are defined as CSCW Requests. A CSCW Request contains a header and other fields including RequestType and RequestParameters. The Request-Type is the type of the operation, and the Request-Parameters contains a specific content and parameter(s) related to the operation request.

CSCW Response

The server should respond to the client request and return the result to the client. If the operation is done successfully, the server should return the correct result to the client and update the client user interface, if necessary, to keep the consistence of the product model between the server and clients. Otherwise, it should notify the client and return the reason and error code to the client. A CSCW Response is an operation response that is defined as a triple: <Header, ResponseType, Response>. The ResponseType is the type of the request, and the Response contains the specific response content. Each Response consists of Status Code indicating the success of the operation and Response Data containing the XML data returned to the client from the server.

IMPLEMENTATION

We built a prototype of Web Service based collaborative design CAD system, WSCAD, in which the kernel modeling system is implemented by modifying and extending the functionalities of our existing solid modeling system GS-CAD (Li *et al.*, 1997; Li, 1997). The server executes all the modeling operations, solves feature constraints, maintains the consistency and validity of the product model and coordinates the clients. In WSCAD, some of the functionalities of the original GS-CAD, in particular for interaction with user interface, are moved to the client. In order to reduce network load and amount of data exchanged between the server and the client, the operations should be executed locally on the client side as much as possible, and only the operations that should/can not be implemented on the client are delivered to the server through the invocation of Web Service. The server and the client are implemented in an object-oriented methodology and they interact with each other by invoking Web Services they published.

Feature Representation

As it can only provide the geometric and topological information of the model but cannot provide the non-geometric information such as the tolerance and the dimension information of the model, the traditional CAD system cannot meet the requirements of the new integrated CAD system. A feature based modeling system provides a way to design a central product model in an integrated environment by capturing design intents at an early stage and to maintain the consistency of the product information during the design, analysis, manufacture and check stages.

A feature is the representation of shape aspects of a product related to design and manufacture and has attributes and engineering purposes. Besides the basic geometric and topological information, it contains non-shape information such as feature intents and constraints. In WSCAD, we represent a feature as follows (Li, 1997):

```
Class Feature {
  FeatureType    type;          // feature type
  EntityID       id;            // feature ID
  String         name           //feature name
  Date           date;          // time feature created
  Status         status;        // feature status
  Constraint     location;      // positioning information of
                                   feature
  Constraint     dimensions;    // dimension information of
                                   feature
  Rule           rules;         //design rules of feature
  Description    description;   // description of feature
  Int            number;        // number of associated
  FeatureNode    associations;  // associating net of
                                   feature
  }
```

Each feature is assigned a unique ID that is maintained internally by the system. The dimension and the positioning information of it are given by imensions and locations, respectively. Design rules such as the tolerance and size of the feature in some applications are stored in rules. The description field is used to describe the engineering information of the feature. The associations field contains other features related to it.

Collaborative Design with Feature

In a distributed multiple-user environment, inconsistency of a product model often occurs when more than one user operates on a model simultaneously, which would not happen in a single-user environment. Consider a scenario, in which a deleted feature could not possibly be edited. This is true under a single-user environment, but would not necessarily happen in a multiple-user environment. If this is not dealt with properly, it may cause serious problems such as inconsistency of data structure and the deadlock resulted from processes waiting for each other. Clearly, some rules that define the level for concurrency operation on model and constrains the designer operation, should be used to prevent these conflicts. A main challenge for defining such rule is that it should not limit design flexibility too much. The server of WSCAD can detect this situation, reject the operation, and notify the client that his operation is not meaningful anymore because the feature has been removed.

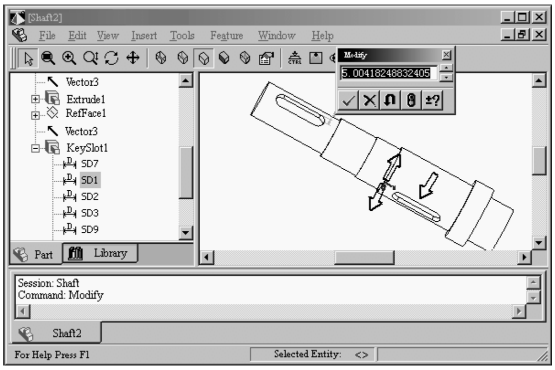
There are three kinds of levels for collaborative design: part level, geometric level and feature level. In part level, before performing modeling operation, a user should lock the model. As the whole model is locked and excluded from other designer, part level for collaborative design provides a very strict concurrency handling policy and can prevent design conflict. However, such level may limit the design flexibility much. On the contrary, geometric level has very freedom for design. In such level, the designer need not lock the whole part before operation, and the server should check any possible conflict caused by collaborative designers. The main disadvantage for such level concurrency is that such loose limit will possibly lower the collaborative design performance and the implementation of it will be very complex.

Comparing these two levels, a good compromise is the feature level concurrent policy. Here, the feature is a representation of the shape aspects of a product that are mappable to a generic shape and functionally significant for some product life-cycle phase. The feature level for collaboration is higher than geometric level and lower than part level. In feature level, the atom of lock is the feature, and a designer should lock a feature before operating it or other feature that associates with it. The designer that locks this feature preempts this feature and other designers cannot operate it until he unlocks it.

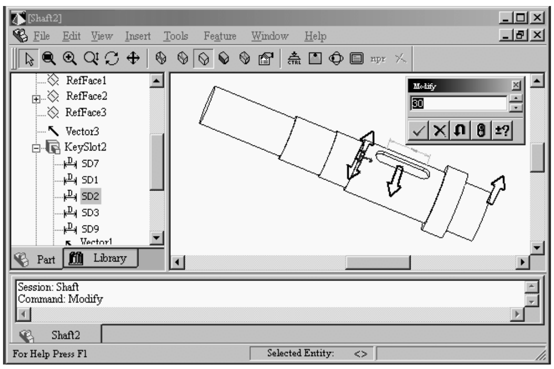
As described in the previous sections, in WSCAD, it is the task of the session manager to coordinate collaborative participants and handle concurrency. Here, a feature is the basic collaborative design element. A designer should lock the feature before operating, and other designers are excluded from operating on it until it is unlocked. Such scheme can prevent the design conflict effectively. Moreover, compared to part level concurrency, feature level would not limit the freedom much. When conflict happens, the design system needs to inform the users related to this conflict about the situation and present some options to solve it. The system should also provide several communication approaches that are used in different conditions for the designers to negotiate. For example, message system, white board and electrical conference can be used by designers to negotiate design conflict.

A Possible Scenario

Our feature-based collaborative design experimental system is demonstrated as follows (Fig.5). Client users log on to the server according to the server's IP and port. After logging onto the server, the client may start a new session or join the current session, and collaboratively design with other participants according to certain rules. Figs.5a and 5b show user1 and use2 modify the depth of keyslot1 and the width of keyslot2 respectively in a design session. After the users set these parameters properly, these operations are handed over to the server for execution and the client user interfaces are updated.



(a)



(b)

Fig.5 Two clients graphic interface of the prototype
(a) User1 modifies the depth of keyslot;
(b) User2 modifies the width of keyslot

CONCLUSIONS AND FURTHER WORK

Web Service, built on the open standard, introduces a new computing concept by enabling different clients to access its service. Based on the analysis of the current collaborative design system, this paper proposes a novel Web Service based distributive collaborative CAD system, employing feature as its collaborative design element. The architecture of the system and the communication protocol used in the system are discussed. The system proposed uses XML for exchanging data between the client and the server. Since operations, such as query and semantic indexing, can be effectively implemented on the XML represented data, it is easy to manage, share and reuse the product model data among users. Moreover, as the related protocol is represented with XML, the system has the virtue of extensibility and platform-independency. Experiment results from a prototype system showed that the system can effectively support the collaborative design under a distributed environment.

Future work should be directed at:

(1) Compressing the geometric model data transferred between the client and the server to reduce network load further.

(2) Security and protection of CAD data. Under the collaborative design environment, these are important issues. Although the current prototype system did not address these issues, further work should focus on them.

ACKNOWLEDGEMENT

The authors sincerely thank Dr. Wang Jianfeng in Tokyo University and Dr. Liu Shengli for their advice in writing the paper.

References

Agrawal, A., Ramani, K., Hoffman, C.M., 2002. CADDAC: Multi-Client Collaborative Shape Design System with Server-Based Geometry Kernel. ASME Design Engineering Technical Conference, (DETC2002/CIE-34465), Montreal, Canada, p.57-65.

Chan, S., Wong, M., Ng, V., 1999. Collaborative solid modeling on the WWW. *In*: Proceeding of the 1999 ACM symposium on Applied Computing, San Antonio, CA, p.598-602.

Heather, K.R. 2001. Web Services Conceptual Architecture (WSCA 1.0) Part I. http://www-900.ibm.com/developerWorks/cn/webservices/ws-wsca/part1/index_eng.shtml.

Hoffmann, C.M., Joan-Arinyo, R., 1998. CAD and the product master model. *Computer-Aided Design*, **30**(11):905-918.

Ip, W.L.R., Lau, W.C.H., 1998. Generic methodology for integration of manufacturing information systems. *International Journal of Computer Applications in Technology*, **11**(6):454-464.

Jian, C.F., Tan, J.R., 2001. Description and Identification of STEP Product Data with XML. *Journal of Computer-aided Design & Computer Graphics*, **13**(11): 983-990 (in Chinese).

Kagan, P., Fischer, A., Bar-Yoseph, P.Z., 2000. Integrated mechanically-based CAE System. *Computer-Aided Design*, **32**(8/9):539-552.

Kao, Y.C., Lin, G.C.I., 1998. Development of a collaborative CAD/CAM system. *Robotics and Computer-Integrated Manufacturing*, **14**:55-68.

Lee, J.Y., Kim, H., Kim, K., 2001. A web-based approach to feature-based modeling in a distributed and collaborative design environment. *Concurrent Engineering: Research and Applications*, **9**(1):74-87.

Lewandowski, S., 1998. Frameworks for component-based client/server computing. *ACM Computing Surveys*, **30**(1):3-27.

Li, H.L., 1997. Research and Implementation of Product Design System in Integrated Environment. Ph.D. Dissertation, Zhejiang University, Hangzhou (in Chinese).

Li, H.L., Dong, J.X., Tang, M., 1997. A Product Modeling System Supporting Feature-Based Parametric Design. *In*: International Conference on Manufacturing Automation (ICMA'97), Hong Kong, April, p.253-259.

Nam, T.J., Wright, D.K., 1998. CoIIIDE: A Shared 3D workspace for CAD. *In*: Proceedings of the 4th International Conference on Network Entities, Leeds, UK, p.103-105.

Rafael, B., Eelco, v.d.B., Willem, F.B., 2001. Collaborative Modeling with Features. *In*: Proceeding of ASME Design Engineering Technical Conferences, Pittsburgh, Pennsylvania, p.1-11.

Rezayat, M., 2000. Knowledge-based product development using XML and KCS. *Computer-Aided Design*, **32**(5-6):299-308.

Rosenman, M., Wang F., 2001. A component agent based open CAD system for collaborative design. *Automation in Construction*, **10**:383-397.

Stock, A., Jasonch, U., 1997. A Collaborative Engineering Environment. *In*: Proceedings of TeamCAD'97 Workshop on Collaborative Design, Altana, GA, p. 25-33.

Szykman, S., Racz, J.W., Bochenek, C., Sriram, R.D., 2000, A web-based system for design artifact modeling. *Design Studies*, **21**(2):145-165.

Wang, H.F., Zhang, Y.L., Cao, J., 2001. Feature-based collaborative design. *Journal of Computer-aided Design & Computer Graphics*, **13**(4):376-372 (in Chinese).

Willem, F.B., Eelco, V.D.B., Rafael, B., Alex, N., 2001. Essential Developments in Feature Modeling. *In*: Proceedings CAD/Graphics 2001, Kunming, China, p.22-24.

Zhuang, Y., Chen, L., Venter, R., 2000. Cyber eye: an internet-enabled environment to support collaborative design. *Concurrent Engineering: Research and Applications*, **8**(3):55-63.