REALTIME ONLINE COLLABORATION FOR COMPUTER-AIDED DESIGN USING A PEER-TO-PEER NETWORK

Hung-Ming Chen[1], and Hung-Chun Tien[2]

ABSTRACT

The Internet has revolutionized the way software systems work. Many applications, such as online games and messengers, have utilized web-related technologies to cooperate over the Internet. Recently, some researchers have utilized web-related technologies in Computer–Aided Design to allow real-time collaboration between users over the web. All these systems are based on the client-server model. In this kind of collaborative design environment, the system grouping, operation and communication all have to rely on the central server, thus the computing loads are concentrated on the server. In addition, the communication becomes less efficient as the clients increase because all clients connect to the server only and all messages must be broadcast by the server using unicast. Furthermore, the availability and stability of the server are crucial.

Considering the real-time collaborative design scenario, its characteristic is all designers in a design team are equal in duties, loadings and functionalities, which is quite similar to the characteristic of all peers in a Peer-to-peer (P2P) network. In order to improve accessibility and flexibility in collaborative design and provide a more load-balanced and extensible environment, this paper presents a prototype implementation of collaborative design based on a P2P model. The users can conveniently form design groups, by connecting directly to each other, anytime and anywhere, without the presence of a central server. All peers are equal in functionalities and computing loads. Based on the P2P network model, several mechanisms are proposed in this paper to form a working model of this system. Then, the object model design and implementation of this P2P system is presented.

KEY WORDS

computer-aided design, Internet, real-time collaboration, distributed processing.

---

[1] Assistant Professor, Dept. of Construction Engineering, National Taiwan University of Science and Technology, 43, Sec. 4, Keelung Rd., Taipei 106, Taiwan, Phone 886/2737-6585, FAX 886/2737-6606, hungming@mail.ntust.edu.tw

[2] Graduate Research Assistant, Dept. of Construction Engineering, National Taiwan University of Science and Technology, 43, Sec. 4, Keelung Rd., Taipei 106, Taiwan

INTRODUCTION

Using Computer-Aided Design (CAD) systems for complex drawing or rigorous 3D model design usually is time-consuming and labor-intensive work. For large-scale design projects, organizing a design team to share the work and cooperate concurrently can speed up the process if the communication between designers and the integration of separated jobs can be well handled. The Internet is the perfect media for group communication and data integration. A collaborative design environment, adopting Information Technology to enable direct communication and data exchange over the Internet between geographically distributed CAD systems, is expected to greatly improve efficiency and convenience for the design group.

Some researchers have utilized web-related technologies in Computer–Aided Design to allow real-time collaboration between users over the web. Abdel-Wahab et al. (1988) established a UNIX-based collaborative environment using Internet and UNIX interprocess communication. Mitchell (1995) pointed out that industrial development and academic research have significantly advanced the technology available for network-enabled CAD applications. Recent system developments include Kao and Lin (1996), Pang and Whittenbrink (1997), Nam and Wright (2001), and Tay and Roy (2003). All these systems successfully showed that geographically distributed users can share data and work together to build and edit visualized models using the Internet.

All these systems are based on the client-server model. In this kind of collaborative design environment, the system grouping, operation and communication all have to rely on the central server. The drawbacks of these systems are computing loads are concentrated on the server and the availability of the server is critical. The rigorous reliance on the central server makes the forming of design groups somewhat inconvenient and inflexible, and any fault on the server will crash the whole group.

Considering the real-time online collaborative design scenario, which is each designer is equally assigned part of the design project and works concurrently, the computational loads would be better distributed to each designer's machine, not concentrated on the central server. In order to improve accessibility and flexibility in real-time collaborative design and provide a more load-balanced and extensible environment, this paper presents a prototype system design and implementation of real-time collaborative design based on the Peer-to-peer (P2P) model. Using the P2P network for collaborative design, the users can conveniently form design groups by connecting directly to each other without the presence of the central server, and all peers are equal in functionalities and computing loads.

For improved serviceability, load-balancing and performance of the current client-server based collaborative design systems, this paper presents a new real-time online collaborative computer-aided design system named ROCCAD based on the P2P network model and multicast technology. The data is communicated on an overlay network formed by a set of unicast (one-to-one) connections between peers. The connection rule is a peer can connect to at most one peer and be connected to by many peers, thus the resulting network is a tree network (Tree First). This Tree First P2P based overlay network is built totally on the application level since it is based on standard Internet protocols and network equipment. It is highly extensible since all the communications are handled by the application program. The concept of this prototype system is shown in Figure 1.

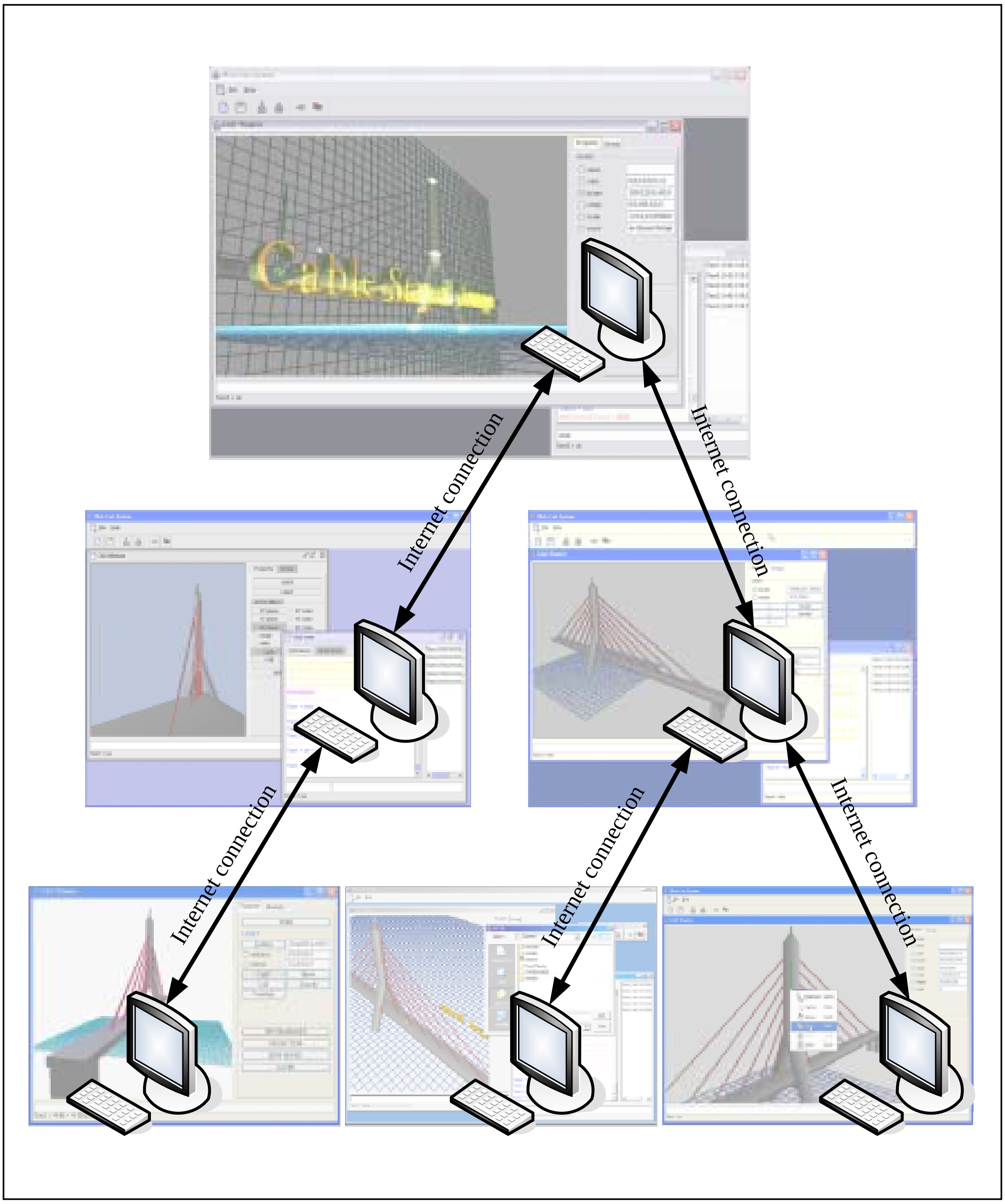

Figure 1: Concept of the proposed prototype system (ROCCAD)

SYSTEM REQUIREMENTS

From a system development perspective, some requirements are crucial for successfully permitting P2P-based real-time online collaboration for computer-aided design. Discussion of these requirements follows with some solutions provided.

COMMUNICATION EFFICIENCY

The real-time collaborative design process involves frequent communications over the Internet among peers. The efficiency of Internet communication is the key to system success. In order to permit every user to view all the users' progresses on a synchronized viewport and communicate with others in real-time through interfaces, the most common communication in the proposed environment is each peer updates its state to the other peers. The characteristic of this communication is an identical update message to all the other peers. Therefore, multicast (one-to-many) technology can be adopted to improve the efficiency in communication. The proposed system adopts the Application Level Multicast (ALM) scheme. ALM is message broadcasting handled by the program at the application level. The application has to be programmed to deal with message passing. The efficiency is improved by activating a message receiver as a new message sender immediately after completing the receiving.

DATA MANAGEMENT AND SYNCHRONIZATION

The collaborative design system needs to maintain and manage design data created by multiple users in a distributed environment. In addition, each peer must have the ability to communicate directly with the other peers over the Internet. Besides the low-level TCP/IP protocol, another high-level mutual protocol on the string messages for message identification and processing is necessary.

In the proposed system, the design data is stored and communicated in the same format, which is a string started by a keyword specifying its type and followed by its individual attributes. For recognition and processing of data in a distributed environment, a common protocol among peers is established by system rule which reserves a keyword and specifies a set of attributes for each type of data. On the other hand, each datum must have a unique ID to be distinguished in the distributed environment. The proposed data labeling scheme to ID design data is its creator's ID plus a serial number, and the ID of a user is the IP address of the user's machine following by the connecting port. This scheme can ensure all the resulting IDs in a distributed environment are identical.

In order to achieve data synchronization so that every user can consistently view the shared model, which is constantly updated by all users in real-time, each peer has to notify all the other peers by broadcasting its modification, such as addition, deleting, scaling, or moving a modeling object, to the model immediately after an action is completed and confirmed. As a result, all peers in a group have the same data all the time to achieve data consistency and synchronization. In addition, any newly joined or rejoined peer must clear its data and then download all the current design data from the peer with which it connects to ensure data consistency.

SYSTEM WORKING MODEL

A set of distributed mechanisms is required to form the system working model of the proposed distributed system for each peer to thoroughly handle all the possible issues and scenarios, such as the joining and leaving of a user, access and request conflicts, data transmission, data consistency and synchronization, etc.

For user's convenience, this system adopts the IP multicast technology to provide the new user a list of IPs and ports of the peers currently in the group for selection to connect to. A new user can request and receive all the IPs and ports of the peers in the group through the router. However, the IP multicast only works among the peers within the same Local Area Network.

The proposed on-line process begins with a peer sending a request to the root peer before joining or leaving the group. Upon receiving the request, the root peer will update its group list by requesting all the other peers to send their information to it, including the newly joined peer. After the root has updated its group list, the list is broadcast to all the other peers and replaces the old list to complete the list update of the group. After the new peer joins and the group lists of all the peers are updated, the new peer can obtain the current design data by simply downloading all the design data from the peer which it connects to.This mechanism is robust and the resulting group lists are in the same order on all peers.

When a peer leaves the design group, other peers must not be affected by its leaving. To achieve this, the Internet connections of those peers connected to a leaving peer must be recovered to reconnect with some other peers in order to keep those peers in the group. A recovering peer will first try to reconnect to the peer which the leaving peer was connected to. If such a peer does not exist, this means the leaving peer is the root. In this case, the peer ordered first in the leaving peers becomes the new root of the design group and the other leaving peers connect to it to recover their connections.

In a collaborative design environment, all the design component objects are openly shared and accessible by all users. To avoid an object being manipulated by more than one user, a lock-based mechanism for object access is necessary. If a user wants to manipulate an object, he or she must request a lock on the object first. After the lock is granted, the requesting user becomes the owner of the object and can modify it. To ensure a lock on an object will not be granted to more than one user, a locked object cannot be requested until its owner releases its lock. To ensure all the locked objects belong to an owner, a peer automatically releases all its locks on objects immediately before leaving the group.

The proposed lock-based mechanism for P2P network begins by broadcasting the locking request from the requesting peer to all the other peers. Upon receiving the request, each peer simple tries to lock the object for the requesting peer. No matter if the locking is successful or not, no confirming or failure message is sent back to the requesting peer for efficiency. Therefore, both locking and unlocking are done after a broadcast request in this mechanism. Although most of the locking requests can be done efficiently in this way, a requesting conflict is possible if another user requests a lock on the same object while a locking request on it is still transmitting and not completed yet. This conflict occurs when any peer who receives multiple locking requests or if the requesting peer receives requests on the same object. To resolve this conflict, the locks of the object from all peers must be released. Because only the owner peer has the right to unlock its objects, the proposed solution is a requesting peer automatically broadcasts an unlocking message to all peers to cancel the locks of those peers which have received its locking request whenever it receives another locking request on the same object which it is requesting.

In order to maintain data consistency and synchronization among all peers, each peer must update its state to the other peers by broadcasting an update message frequently. To broadcast an update of a peer, each peer forwards a received message upward or downward to its connecting peers which have not received the message yet by following the tree network structure. Since all peers have to send their own updates and forward the updates received from connecting peers concurrently, it is very difficult to schedule these communications through a single port. A feasible solution is every peer has to create multiple connections through different ports for each of its connecting peers to form an overlay network and handle the communications with all its connecting peers concurrently. The multi-thread capability included in Java can be used to fulfill this goal by creating multiple threads to run concurrently and each thread has a connection to another peer.

In addition to a data transmission mechanism, a set of update frequencies is necessary for each kind of message to determine how often a peer should broadcast its various updates. The update frequency of the design data is a peer sends out its update whenever the user adds an object, deletes an object, or releases the lock on an object. The update frequency of the message board is a peer sends out its update whenever the user hits the "enter" key. The update frequency of the whiteboard is a peer sends out its update whenever the user draws a new line.

USER INTERFACE

A 3-dimensional (3D) design environment with interactive and graphical user interface to support CAD operations is the basic requirement for modern computer-aided design systems, and also is basic for the proposed collaborative design system. Therefore, a viewport, for each peer, for synchronously displaying the whole 3D design project worked on by the design team in the shared virtual design space is necessary. The current design model should be visually displayed as 3D graphics in the viewport using a 3D graphics library.

In addition to the graphical interface for displaying the shared design project synchronously, communication interfaces can further facilitate cooperation by allowing users to communicate in real-time through the system directly. Therefore, the proposed system should provide a message board and a whiteboard for Internet conferencing.

PLATFORM INDEPENDENCE

Any eligible user must be able to join a design team without platform limitation due to user's machine. Therefore, computer collaboration between heterogeneous machines is necessary.

Because platform-independence is a requirement, a platform-independent programming language, Java, was the choice for implementation of the proposed system.

COMPATIBILITY

The proposed system should have the ability to be compatible with popular commercial CAD systems to improve the system serviceability. The proposed solution is that design data can be transferred between a commercial CAD system and the proposed system through file import and export of the proposed system. The proposed system is on top of commercial CAD systems to form the collaborative design environment. The user can use a commercial CAD system under the proposed system to do the design. This integration can allow users to perform design more conveniently and flexibly. The proposed system can take advantage of the functions provided in commercial CAD systems.

SECURITY

The proposed system facilitates collaboration via the Internet. As a web-based distributed system, it is quite possible for the system to be harmed by a computer virus or hacker attack. To maintain a stable and secure collaborative design environment, network security must be appropriately addressed. Therefore, password login checks, firewalls and anti-virus scans are mandatory for system protection.

IMPLEMENTATION

The proposed prototype system can be thought of as the fusion of a CAD program and a messenger program. Considering the functionalities of these two programs, the proposed system architecture is a three-tier system. The first component is the CADSystem which provides CAD functionalities and an interface to aid the user to do the design and synchronously view the progress of the collaborative design project worked on by multiple users in real-time. The second component is the MessengerSystem which provides messenger functionalities and interfaces to allow the user to do Internet conferencing with the other peers. The third component is the WebCommunicationSystem which provides functionalities to allow the proposed system to connect and communicate with each peer over the Internet.

The object model of the proposed prototype system (ROCCAD) is shown in Figure 2. The WebBasedCadSystem class represents the web-based windows interface which contains a CAD window and a messenger window and has the ability to cooperate over the web. The CADSystem class, the MessengerSystem class, and the WebCommunicationSystem class represent the three major components of the proposed system architecture respectively. The FindServerDialog class represents the window dialog which will pop out for user's inputs while processing the user's online request.

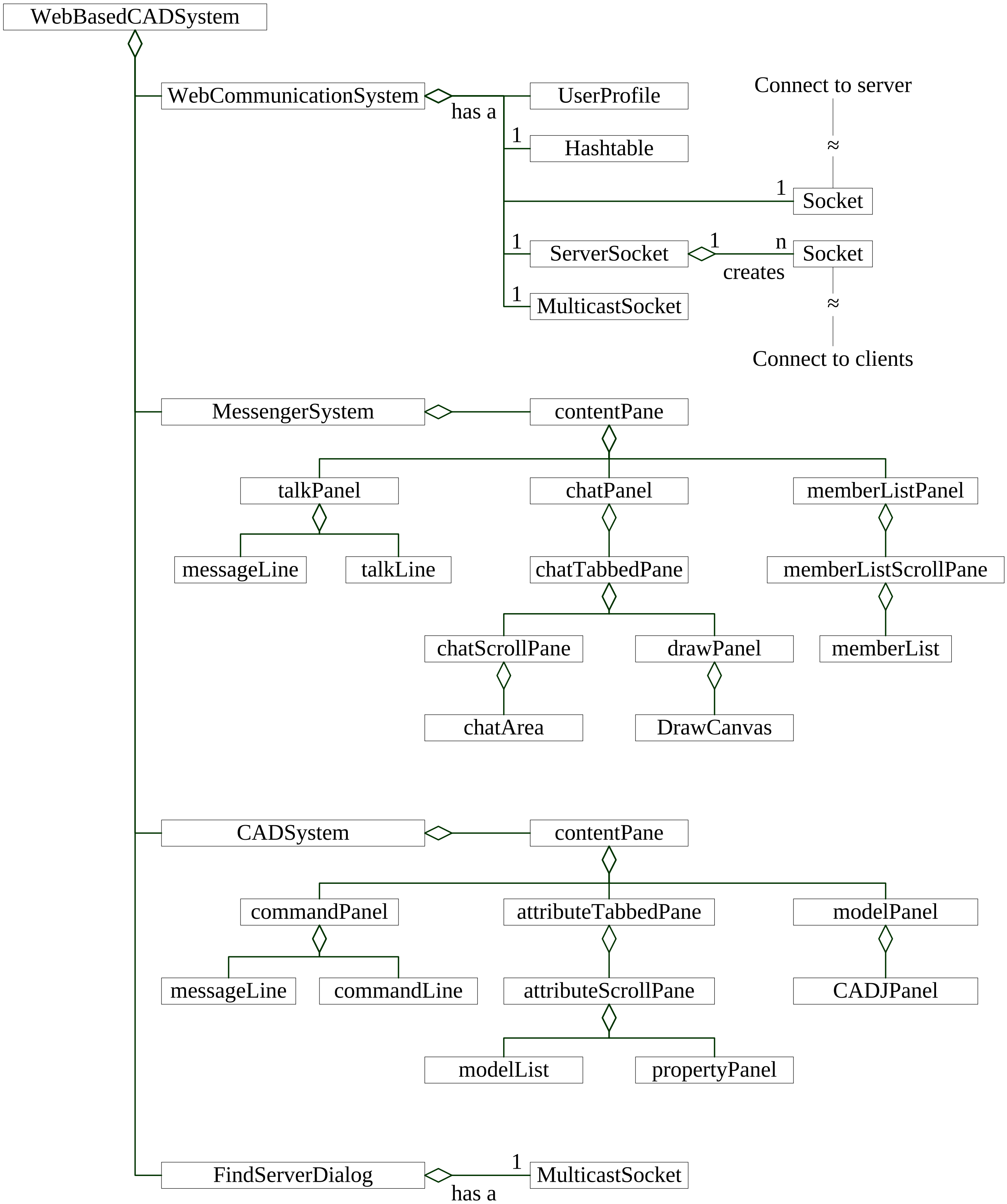

Figure 2: The object model of the proposed prototype system (ROCCAD)

The WebCommunicationSystem Class handles all the communication-related tasks of the proposed system, such as connections and communications with other peers over the Internet, based on the proposed mechanisms. The UserProfile class is for storing the information, such as the name, IP, and port number, about a user in the design team. All UserProfile objects together form the group list. The Hashtable class is for storing the communication-related information about all the peers connected to it. The conceptual Internet connection in Java programming is between the built-in Socket objects, provided in the java.net package, on both sides. Each Socket object has an Address object to record the unique IP address of the local machine for identification on the Internet. A ServerSocket object can passively establish Internet connection with many remote Socket objects by constantly listening to the Internet for connection requests from remote Socket objects. If a connection request is accepted by the ServerSocket, it creates a local Socket object to connect with the remote Socket which sent the request, then returns to the listening state and waits for the next connection request. The MulticastSocket class provides IP multicast functionality for looking up the IPs of the peers in the group.

The MessengerSystem class represents the communication window of the proposed system. It provides communication interfaces, such as the message board and whiteboard, and handles all conference-related tasks with help from the WebCommunicationSystem.

The CADSystem class represents the CAD window of the proposed system. It provides CAD interfaces and handles all CAD-related tasks, such as lock-based object accessing, based on the proposed mechanism with help from the WebCommunicationSystem. The CADJPanel is the main 3D graphics class of the proposed system. It handles the 3D rendering output on the viewport based on the current design data and environment settings using OpenGL, the processing of various events, the saving and transformation of modeling data, and the control of popup menus.

CONCLUSIONS

This paper has presented a prototype system, named ROCCAD, for real-time online collaborative computer-aided design. Compared with the current client-server based collaborative design systems, the features of this system are that online collaboration is based on a P2P network for improving the convenience and flexibility of system usage, and the communication is based on the Application Level Multicast (ALM) scheme for improving efficiency. After taking into consideration various aspects of implementing this P2P based system, the system requirements have been addressed. To fulfill these system requirements, a set of mechanisms has been proposed to form the system working model of the system. Then the object model design and implementation of ROCCAD have been presented.

Considering the real-time collaborative design scenario, its characteristic is all designers in a design team are equal in duties, loadings and functionalities, which is quite similar to the characteristic of all peers in a P2P network. Therefore, using a P2P network is more natural and load-balanced for real-time collaborative design. In addition, the P2P model has several advantages from a user’s view point in comparison with the client-server model for collaborative design. First, it is more convenient since the users can form a design group by connecting directly to each other anytime and anywhere without the presence of a central server. Second, it is more flexible because any peer is allowed to join or leave the network during the design process without limitation. Third, it is more robust because the malfunction of any peer will not affect the other peers as well as the whole design project. Fourth, it is more extensible because of the decentralized network and the ALM scheme. Although the system working model in a P2P network is somewhat complex, it has been proved that the proposed mechanisms can conquer the complexities and achieve the same functionalities for supporting real-time collaborative design as using the client-server model.

ACKNOWLEDGMENTS

This research was supported by the National Science Council of Taiwan under Project Number NSC 93-2211-E-011-018.

REFERENCES

Abdel-Wahab, H. M., Guan, S. -U. and Nievergelt, J. (1988). "Shared workspaces for group collaboration: an experiment using Internet and UNIX interprocess communications." *IEEE Communication Magazine*, 26(11), 10–16.

Kao, Y. C. and Lin, G. C. I. (1996). "CAD/CAM collaboration and remote machining." *Journal of Computer- Integrated Manufacturing Systems*, 9(3), 149–160.

Mitchell, W. J. (1995). "CAD as a social process." CAD futures 1995, *Proceedings of the 7th International Conference on Computer-Aided Architectural Design Futures*, Munich, Germany, 7–9.

Nam, T. -J. and Wright, D. (2001). "The development and evaluation of Syco3D: A real-time collaborative 3D CAD system." *Design Studies*, 22(6), 557–582.

Pang, A. and Wittenbrink, C. (1997). "Collaborative 3D visualization with CSpray." *IEEE Computer Graphics and Applications*, 17(2), 32–41.

Tay, F. E. H. and Roy, A. (2003). "CyberCAD: a collaborative approach in 3D-CAD technology in a multimedia-supported environment." *Computers in Industry*, 52(2), 127–145.