Joint Conference of 1996 International Computer Symposium
December 19~21, Kaohsiung, Taiwan, R.O.C.

Study and Implementation of Integrated CSCW Support Environment*

Yong Wang Xianji Li Yong Tang

Member of Computer Society of Guangdong Member of Computer Society of Guangdong Member of Computer Society of Guangdong

(E-mail: yongwang@gdut.edu.cn, lixianji@gdut.edu.cn, ytang@gdut.edu.cn)

Abstract

This paper proposed a new solution to construct an Integrated Computer Supported Cooperative Work(CSCW) Support Environment, and introduced the architecture of the Integrated Environment, message and hook, designing and implementing technologies of Integrated Environment and relative problems as well.

1 Motivation

CSCW(Computer Supported Cooperative Work) is a new research area. It lays stress on studying characteristics of cooperative work and architecture of cooperative work systems, etc.. CSCW is benefit to users to communicate with and cooperate each other. It provides some functions which support coordination and cooperation when members of workgroup are sharing work, and it has great efficient.

Now, there are some products about CSCW which may have high performance and overall functions, but they need much longer time to develop and are more expensive too. On the other hand, there are a lot of application programs which are provided by Windows. These Windows application programs are with many functions and high performance too. They have been tested by market critically, and supported by many users and applied widely. How to inherit this huge kind of wealth and how to use these application programs to develop and apply to CSCW systems are a very significant task. Our task is exactly to do so.

*This research is supported by the Education Department of Guangdong Province and Guangdong Provincial Natural Science Foundation of China.

2 Architecture of Integrated CSCW Support Environment

2.1 Principle of Integrated Mechanism

Our Integrated CSCW Support environment is such the one in which many original Windows application programs without any modification can be run and they can be used for doing cooperative work wonderfully, thus cooperative editing, painting and discussing through networks by members of a workgroup can be implemented. Therefore, we can develop and expand functions of CSCW systems at a less cost by inheriting and using original Windows application programs.

2.1.1 Message

Windows is a multitasking operating system, one of kernel characteristics of Windows is called a message. Any Windows program is driven by messages. Any event of the system such as a mouse motion, a keyboard press and hardware interrupts and so on, can bring about messages. Message is actually a kind of data structure, it registers

message ID and some other additional information about this event. At first, messages are arrayed in the system message queue, and then, Windows system sends messages in the system queue to relative application message queues, the message loops of the applications will get message in their message queues, and notice the system to send the messages to relative windows. This is the message mechanism (see Fig. 1).

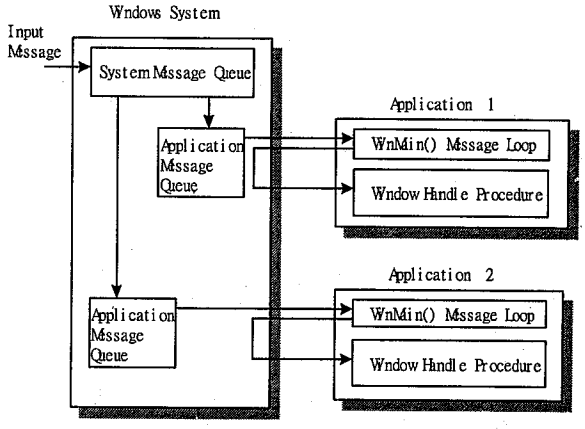

Fig. 1 Message

2.1.2 Hook

Hook is one part of Windows message mechanism, it provides a feature to intercept and modify messages in Windows system. Hook Callback Functions can be inserted into message flows to intercept and lookout messages, and handle messages before they arrive at their destination windows.

Windows system offers many varied kinds of hooks: system queue hook, message transmission hook, message filter hook, journal hook, Computer Based Training(CBT), and etc.. Every kind of hook provides one way to handle messages in Windows. For example, application programs can use mouse hook(WH_MOUSEMOVE) to lookout mouse motion messages.

Generally, hooks have two kinds of action areas, system and task. Hooks in System action area are effective to every application in system, but hooks in task action area are only effective to some special applications. Some kinds of hooks are in two action areas, others only in system action area. On the other hand, some kinds of hooks can modify messages, some can be only used to filter messages. Applications should be designed and programmed by selecting suitable hooks and action areas according to their application targets and demands so that system will have higher performance.

Windows manages and dispatches hooks by managing and maintaining an independent hook chain for every kind of hooks which is a table pointing to Hook Callback Functions. When a special message which is relative to certain kind of hook emerges, the system transmits the message to every Hook Callback Function in hook chain one by one, at last, it tansimites message to destination windows processing procedure. Functions of Hook Callback Functions will be decided by the kind of hooks.

2.2 Architecture of Integrated System

The idea of integration is based on the message mechanism so that messages can be transmitted to other computers in network and drive applications on other computers as well as on the local computer, that can lead to cooperative work. Because this kind of cooperative work is achieved by messages driven, it is relatively independent to application programs, Windows applications can be being shared applied and cooperative working without modification. We can construct Integrated CSCW Support Environment by using message mechanism.

Based on message mechanism and hook, we studied and implemented an Integrated CSCW Supported Environment. In this environment, hooks are applied to incept messages; and mailslots are used to transmit messages. The architecture of Integrated Environment is shown in Fig. 2.

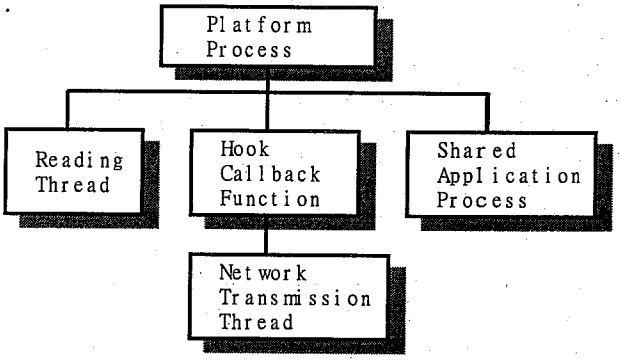

Fig. 2 Architecture of System

3 Implementation and Application

3.1 Software Architecture

We have programmed the application program called PLAT.EXE in Visual C++2.0 which is a platform of Integrated CSCW Support Environment. PLAT can support and coordinate Windows applications in CSCW system, and it will achieve initiation and launch applications, and set hooks and transmit messages through network, etc.. For high performance, we applied

multitasking architecture which is made up with multiprocessing and multithread in Windows 95. So we excavated corporation fully to increase performance of system.

PLAT, the platform program ,and applications are combined of multi-processes. We designed reading thread that is due to read messages from network and send these messages to relative applications. Reading thread and Hook Callback Functions are composed of multithread architecture. Fig. 3 is the software architecture of Integrated Environment.

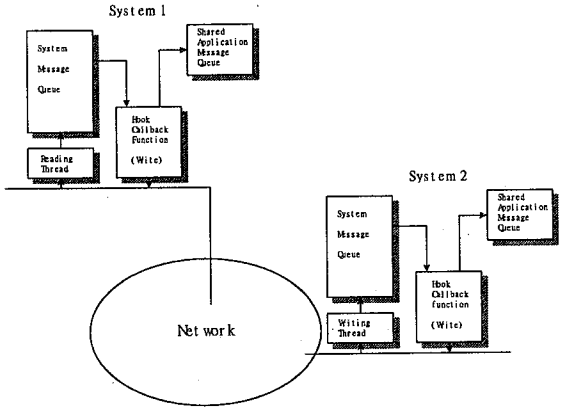

Fig. 3 Software Architecture

3.2 Implementation

There are many kinds of messages in Windows, so many Hook Callback Functions for messages should be programmed inside the Integrated Environment to process many kinds of messages.

On the other hand, Integrated Environment should provide support to many applications, so Hook Callback Functions should be shared resources, and programmed inside Dynamic Link Library(DLL).We programmed a DLL called CSCWHOOK.DLL to offer many kinds of Hook Callback Functions to Integrated Environment. Meanwhile, since hooks are for applications, it will have high performance only in task action area.

In PLAT we used the obvious method to link DLL, and used procedure LoadLibrary() to load CSCWHOOK.DLL into memory. And then we used SetWindowsHookEx() procedure to install suitable Hook Callback Functions to applications. When system needs no DLL, platform would use FreeLibrary() to free pure memory resource immediately.

We used CreateProcess() to launch a application, and after this procedure returned the handler of the application, SetWindowsHookEx() used it to set Hook Callback Functions to this application.

Network transmitting is aided by mailslot which is supported by Windows 95. Mailslot is a simple method to transmit messages through network, and it has two kinds of application modes, one is peer to peer, the other is broadcast. Broadcast can support multicast cooperative work.

The Integrated Environment is composed of three parts: PLAT.EXE, the platform of Integrated Environment; CSCWHOOK.DLL which provides many Hook Callback Functions; and shared applications. When the Integrated CSCW Support Environment is running, users can apply by control menu, for example, people can select shared application by pplication" dialog through pplication" bars in control menu. After an application was selected, Hook Callback Functions were installed automatically.

4 Discussions

At the beginning of this task, we designed a demo program, DEMO.EXE, which executed paint function as well as MSPaint.exe in Windows 95 so as to help and test developing of this Integrated Environment. During DEMO.EXE run, users of workgroup could not operate at same time, otherwise, some additional lines will be drawn in the guest window of DEMO.EXE, as a result, cooperative work is influenced seriously. The reason is that messages which come from different computers are arrayed in one system message queue so that Windows application can not distinguish between messages from local computer and messages from remote computers. Because coordinate data in different computers in network are different, application will be confused and draw lines among different coordinate data additionally.

There are two solutions: one is designing special application for CSCW, the other is using synchronization and mutual exclusion. We examined two methods above to realize CSCW applying.

1) Modifying DEMO.EXE

We added a new kind of messages to indicate messages from different computers. For example, if computer received a mouse motion message, WM_MOUSEMOVE, from network, reading thread will translate the message to ID1_MOUSEMOVE message. We modified DEMO, and added program segments to process ID1_MOUSEMOVE and other new message IDs. So the application will be able to distinguish messages from different computers and process correctly.

This method needs to modify sources of applications, this is very limited at this point. So it is not so good for our Integrated Environment, and we examined another method

at this meantime.

2) Adding synchronization and mutual exclusion to PLAT.EXE

Because messages come from other different computers and messages from local computer are processed by different threads, reading thread and transmitting thread. So we can apply synchronization and mutual exclusion between two threads, and treat messages queue with critical resource.
This method avoids messages from different computers being arrayed alternately, and need not modify application programs. But it limits concurrently operating at same time, users must do cooperative work one by one although this is a effective mode for CSCW applying.

5 Conclusion

The Integrated CSCW Support Environment is a good platform for Windows applications. We have achieved our target, and now we are developing video and audio functions. Next, audio/video and Integrated Environment will be combined to form the base for developing video conferencing systems, remote education systems, remote treatment systems, and so on. These systems are typical applications of CSCW.
CSCW is the very important research area in recently computing technologies. Developing the Integrated CSCW Support Environment will be a good way to study CSCW.

References

[1]Roddwn T.A. Survey of CSCW systems. *Interaction with Computer*,1991,3(3).
[2]K.Watabe,S.Sakata. Distributed desktop conferencing system with multiuser multimedia interface. *IEEE Journal Selected Areas in Comm*, 1991, 9(4):531-539.
[3]Meilin Shi. CSCW: Computer Supported Cooperative Work. *Development and Application of MicroMini Computer,1995,4.*
[4]CROWSTON K., MALONE T.W. Computational agents to support cooperative work. *Working Paper No.2008-88, Center for Information Systems Research, Massachusetts Institute of Technology, Cambridge, Mass.*. 1988
[5]Greif, I. Computer-supported cooperative work: Break-throughs for user acceptance. In *Proceedings of the ACM Conference on Human Factors in Computing Systems. ACM, New York.* 1988:113-114.
[6]Robinson, Mike. Computer Supported Co-operative Work: Cases and Concepts. In *Proceeding of Groupware'91, UL:Software Engineering Research Center.* 1991.
[7]Schmidt, Kjeld. The Articulation of Cooperative Work-Requirements for Computer Support. In *Developing CSCW Systems: Design Concepts, ed K. Schmidt. Report of COST14 oTech' Working Group 4.* 1993.
[8]Malone, T. W., Grant, K. R., Lai, K. Y. The Information Lens: An intelligent system for information sharing and coordination. In *Technological Support for Work Group Collaboration, M. H. Olson, Ed. Lawrence Erlbaum, Hillsdale, N.J.* 1989.