M.Tech. (Computer Science) dissertation Series

PROTECTED BULLETINE BOARD

a dissertation submitted in partial fulfilment of the requirements for the M.Tech. (Computer Science) degree of the INDIAN STATISTICAL INSTITUTE

By
P.Balakrishna Reddy

under the supervision of
Prof. Dr. Probal Sengupta
(Computer Vision and Pattern Recognition Unit)

INDIAN STATISTICAL INSTITUTE
203, Barrackpore Trunk Road,
Calcutta - 700 035

31st July , 1998

CERTIFICATE OF APPROVAL

This is to certify that the dissertation titled **PROTECTED BULLETINE BOARD** submitted by **Mr P.BalaKrishna Reddy** to the **Computer Vision and Pattern Recognition Unit, Indian Statistical Institute, Calcutta,** is a bonafide record of the work and investigation carried out by him under my supervision and guidence.

Date: 31st July, 1998
Indian Statistical Institute
Calcutta.

(Probal Sengupta)
Professor
Computer Vision and Pattern Recognition Unit
Indian Statistical Institute
Calcutta

ABSTRACT

The developement of protected Bulletin Board usingTCP/IP is considered ,consisting of a number of users. A bulletin Board can be created from any client program running on machinesin the net work byusers giving protection access to the board and to its page and for the comments generated later on by the users to the pages for which they have access to comment.Once creation of bulletin board is over this will be stored in a machine in the network on which PBBserver(ProtectedBulletine Board) is executing, from which the other users can see the board according to the access rights given by the Owner of the board.

A Bulletin Board is a bunch of information about something with different fields(pages).Each page will give some information about the object for which the board is created.Each page will have access rights which will be given by the Owner(or)Creator of the board.If any user wants to see the Board he or she will be able to see only those pages for which he/she has got comment access(read or Comment). If a user has got comment access then he can see the comments generated by the users provided owners gave access to this user and he can also generate a comment and give access rights to the page,which he/she has generated.

ACKNOWLEDGEMENTS

First of all, Iwould like to thank my Project Coordinator and Guide Prof .Dr.Probal Sengupta, who initiated me into this project. During the development and implementation stages, his timely issuance of approaches,ideas and lessons have helpaed me tosimplyfy the development to great extent.

I thank Prof .Dr. Probal Sengupta for providing me the facilities to implement the programs and perform simulations on the machines in the CVPR unit.

I thank the Staff and Research Scholars of the unit for providing me all help in many different ways.

I specially thank my classmates Sudhakar Rao ,Surekha and my friend Jyosthna for their good support throught the two maddening years .

I thank Parveen Gupta , Bani Kumar Maity,H.K patel, Rajeev Verma for providing resources to complete this dissertation.

Lastly I thank all of my classmates for their cooperation .

P.BalaKrishna Reddy
July 31 1998

CONTENTS

Chapter 1

Introduction

The title of the dissertation is "Protected Bulletine Board". Here we will discuss about What is meant by Protected bulletine board?. First of all What is meant by Bulletine Board? What is protection to that?.

1.1 What is a Bulletine Board?

A Bulletin Board is a bunch of information about something with different fields(pages).Each page will give some information about the object for which the board is created.Each paghe will have an option for the comments which will be given by the users.

1.2 What is protection?

All parts of the Board can't be shown to all users. For example if a Research Scholar wants to create a board about him self.Then he will be giving different types of information about him

For Example

Board

Page 1. Educational details .
Page 2. Papers published .
Page 3. Sports activities .
Page 4. About Personal life .
Page 5. ExtraCurricular activities .

If Owner don't want to show all of his pages to all the users. He can give access to only some users to see his Board Pages. At same time he can give access to some users to comment on his particular page.Similarly all the users who has got access to comment can give comments on the particular page for which they have access to comment at the same time they too can restrict their comments by giving protection access. If any user doesn't have read access then he can't see the page contents.Similarly Comments and he can't comment. This way the Board is protected.

Each page will have access rights which will be given by the Owner(or)Creator of the board.If any user wants to see the Board he or she will be able to see only those pages for which he/she has got access(read or Comment). If a user has got comment access then he can see the comments generated by the users provided they gave access to this user and he can also generate a comment and give access rights to the page,which he/she has generated.

We first discuss about the model that is being followed throught the project, which is Client -Server Model.

The standard model fo Network applications is the Client -Server Model. A server is a process that is waiting to be contacted by a client process so that the server can do something for the client.Atypical(but not mandatory)scenario is as follows:

The server process is started on some computer system. It initializes it self ,then goes to sleep waiting for a client proccess to contact it requesting some service.

A client process is started, either on the same system or another system that is connected to the server system through network. Client Processes are often initiated by an interactive user entering a command to a time sharing system. The client process sends a request across the network to the server requesting service of some form.Some examples of the type of service that a server can provide are:

1. Return the date and time of day to the client.
2. Print a file on a printer for the client.
3. Read or write a file on the servers system for the client.
4. Allow the client to login to the server's system.
5. Execute a command for the client on the srver's system.

When the server process has finished providing its service to the client, the server goes back to sleep,waiting for the next client request to arrive.

Chapter 2

2.1 Problem Specification

Title of the dissertation is "PROTECTED BULLETINE BOARD".

Suppose we have n users Let us say A1,A2,..........,An on a machine or on different machines with client program on all machines and a server programme running on any of these machines or on an entirely different machine .

Client should be able to

1. connect to the server running on any one of those machines or the entirely different machine using aport on which the server is Listening and the IP address of the machine in the network.
2. create a bulletine Board with number of pages and asking for protection for each page which will be decided by the creator of the Board.
3. send whole information to the server.
4. ask the server for the list of boards available on the server
5. List the Boards, show to the users in a ListBox
6. Send a request for a particular board
7. receive it and show the information in the form of board according to the access flags set by server .
8. check for modifications done by the user accordingly send the message to server to save the modifications and comments added by the users.
9. delete the Board

Server should be able to

1. Accept the Connection request sent by the clients.Create a socket to communicate with the client.
2. receive the various messages sent by the client for example messages for DIR,SENDBOARD,SAVE,MODIFY,DELETE etc.Proccess these messages as follows
   - 2.1 On DIR send the List of Boards available in the server.
   - 2.2 On SENDBOARD send the particular board asked by the client with this message

2.3 On SAVE save the Board created by the client and which is sent to the server with this message

2.4On MODIFY save the Board sent by the client which is already in the server and modified by the client

2.5 On DELETE delete the board in server which is specified b the client with this message.

3. whenever client asks for aparticular Board it sets the access flags according to the access rights given by the creator of the board .similarly for the comments according to the access rights given by the users for their comments.

So whole application (Server and Client) will allow user to create a bulletine board from any machine or System. Store it in the server system.and allow any users to see any board according to the access rights given by the creator for the pages and access rights given by the users for their comments.

Chapter 3

Problem Analysis

The first step towards the development of the package is to understand the situation. In order to do so,we first talk about client -Server model and explain how the given problem can be fitted with this model.

By now we have realised that the problem consists of two basic parts one is server the other is client.We will understand details in each side in turn.Starting with the Server side ,it has to respond to the messages sent by the client and send required data from the database of its own which is developed by the server it self with the data sent by the various clients in the form of their boards.

Now we move on to the Client side,it has to give a nice user interface for the creation of the board which will give a feel of bulletine board and then ,it has to collect all the information and send it to the server. Whenever it gets a message from server it has to show it to user with an appropriate User Interface.

3.1 Client -Server Model

A powerful and versatile computing methodology,client-server architecture supports the partitioning of application programs in to logically discrete entities. Typically ,client-server applications comprise three components -User interface,datamanagement and application logic.

The terms Client and Server specify the side of the proccessing split on which the given component functions. The client Component requests services from server components. The server accommodates its clients .Usually, the server listens for a client which establishes a connectivity with it. Then the client sends a query to the server which proccesses the query and sends the results back to the clients.

Traditional Mainframe applications are not client -server because the entire application resides on the mainframe.None of the application logic is performed(or proccessed) on the user terminals. Similarly, traditional personal computer applications such as those running on networks,are not client-server because all proccessing occurs at workstations. Client -Server is combination of Mainfram and personal computer technologies,leveraging the strengths of each environment.

In our Problem the server is passive and will be listening on for clients to establish connectivity,receiving requests from online clients and sending responses to them.

3.2 Tasks of Server and Client

We could realize from our modelling in the previous section, that the problem basicallyconsists of implementing two sides, viz theProtectedBulletineBoardServer(PBBServer) side and the Client side.Each of these sides henceforth will be dealt with, seperately. The order of dealing with these will depend upon the convenience of understanding.

PBBServer will be Listening to the client requests for the connection establishment.Once this process is over Server will create a separate socket for this client and will start waiting for the requestss from clients.The various requests it has to do are like Send List of Boards ,Send the particularBoard, Store the Board in the server, Modified the board by the client save it in modified form,delete the Board etc.Apart from this it has to maintain the list of clients connected to it and the socket communication channel allocated for each client to get and send messages.

Client is active proccess , which will initiate the connection with the server.Once the connection is established it can do various jobs like ask for the BoardsList on the server and send the board, save the modified board,delete Board etc.

Chapter 4

PROPOSED SOLUTION

In this Chapter ,We begin with the way the program is organised .then we would talk about the datastructures, involved, the message passing mechanim,a detailed description of each component of the program.

We have an PBBServer and several Clients. Client will have two components

1. Document and its message handlers
2. View and its message handlers

When the PBBserver starts it opens a listening socket and listens for the clients connection requests on particular port which we specified at the beginning of server on the server machine.

Whenever a client begins and requests for connection by specifying the IPaddress of the server machine and the port on which the server is listeing,server creates a socket and connects the client to this socket for communication with the server.This is also called creating a communication channel .Once this proccess is complete ,listening socket will go for its duty of listening for connection requests from clients.So in this way server will be creating a communication channel for each client connection request. Now for each communication channel server notes the port and IPaddress of the Client for further communication from server side. Whenever an error occurs in recognising the IPaddress or port on which the server is running it will be notified to the client in the form of dialogbox asking for change of address or port.

Once the communication channel is established between the client and server its natural that the client will request for services the server will process the request and sends the information back to the client.

Primary tasks of the PBBServer are the following whenever the client asks for it.

1. Send the list of Boards available in the server to client.
2. Send the particular Board requested by the client after setting the access rights flag.
3. Save the modified Board sent by the client.
4. Once the modified board us saved send the acknowledgement to the client about the modified board received.
5. Save the Board created by the client.
6. Send acknowledgement to the client after the board is saved on this machine
7. Delete the board after checking the password sent by the client to delete the board created by that client.

Last and Final job for the PBBServer is it has to store all the information in the permanent storage like hard disks.

Prmary tasks of the client are

1. It has to send the appropriate requests to the sever according to the user input
2. It will request the client for List of boards,a particular board,Save the created board,save the modified board,delete a board in server, send request for myboards(boards created by that client).
3. Show the list of boards in list box to the user allow the user to select a single board from list of boards.
4. Send request for the particular board which is selected in the List Box
5. Show the Board data which is sent by the server in the form of a bulletine Board(Discussed already).
6. Set the modified flag to the board if the board is modified and send the message to save this modified board .

7. Allow the user to see the comments if he has comment access which will be set by the server according to the accesslist specified by the Creator of the board.
8. Display the system messages sent by the server.
9. Allow the user to modify the Board if he has modifyaccess

First I would like to Discuss about the way the client gets connected to server .

Start the serveron some machine in the network on some available port by specifying the port number.Now the server has started its Listening (Waiting for connection request) for connection requests from clients on port number we specified. To function as specified above we have to create a socket which will be waiting for Connection request. We derived a class called ListenSock from MFC Class CSocket which will enclose the socket functionality will do the above job of listening.

Start the Client On some machine on the network it will ask for the address and port on which the server is running(waiting for client request) if it is able to catch the server on that port in the specified address then the client gets connected to the server and displays a window . If the client fails to connect then asks for retry and then change of address and port. Here we derive a class ConnectSocket from MFC Class CSocket which will enclose the socket functionality will do the job for us.

Commmunication is nothing but sending Messages to the server and recieving the messages. For this we need a well defined datastructure for message.After completion of communication part we have to see the format of the bulletine Board and datastructure required for it .

Now we talk about the data structures used to

1. Communicate Between server and client.
2. Collect information for a Board when it is created.Let us call it Board data.
3. the List of Boards.

Before Discussing about the datastructures I would like to tell you the way Client and server programs are designed. It is completely Object Oriented Design.Every thing in both client and server programs belongs to one or the other classes used in the program. So the datstructure it self is a class itself or belongs to one of the classes.

4.1 DATA STRUCTURES USED

1. Datastructure or Class used to Communicate between client and server.

I have created a class CMsg which is derived from MFC Class CObject

The Contents or data In that class are

1. loginID of user
2. Message or Command (This specifies what sort of service the server has to do for the client)
3. Linked List of Boards
4. Password of aparticular Board to be deleted
5. Board
6. SystemMessage

```
class CMsg
    {
       CString  ID
       CString Command;
       CStringList BoardList
       CString Password
        CBoard *Board  (Discussed later in next page)
        CString SystemMessage
      }
```

DataStrucure to store the Board

We have discussed the format of the Board previously.Here I will give you detailed contents in the board

BOARD

1. Owner
2. Time of creation
3. Time of Modification
4. Password for this Board
5. List of users having Edit Access
6. List of PAGES it contains

    PAGE

    1. Owner
    2. Time of Creation
    3. Time of Modification
    4. Contents
    5. List of users having readaccess
    6. List of users having Comment access
    7. List of COMMENTS (Added by the Owner or users having Comment access)

        COMMENT

        1. Owner
        2. Contents
        3. Time of Creation
        4. Time of Modification
        5. List of users Having read access
        6. List of users having ModifyAccess

The class "CBoard" will have the following data in it

```
class CBoard
      {
```

```
                              CString  Owner;
                              CString CreateTime
                              CString ModifyTime
                              CString Password
                              CStringList Edit Accesslist
                              Head *CPage   (Linked List of PAGE
Objects)
                             }


                         class CPage
                             {
                               CString Owner
                               CString CreateTime
                               CString ModifyTime
                               CStringList ReadAccessList
                                CStringList  CommentAccessList
                               CString Contents
                                Cpage * Next;( for linked list)
                                Head  *Commentdatapage   (Head of
Linked list of  Comments)
                               }


                        class CommentDataPage
                              {
                                CString Owner
                                 CString CreateTime
                                 CString Modifyrime
                                 CStringList ModifyAccessList
                                 CStringList ReadAccessList
                                 CString Commentdata
                                 CommentDataPage * Next(For Linked
List)

                                }
```

CString is an string Object which stores a string
CStringList is an Linked List Object of strings.
For more Inforation see MFC classes.(VC++ ONLIneHelp)

1. DataStructure to store List of boards.

A linked list is maintained to store the Names of boards which are strings MFC object Cstringlist is used store List of Boards.

Picture of the scenario

4.2Server Algorithm

1. Ask for the port on which the server will listen for connection request from clients if it is a valid port display the window else give error message ask for change of port. The user interface here is Dialog boxes.Now the server is ready with a ListeningSocket which will be waiting for client connection requests.
2. ListeningSocket will create a Communication socket when the client request has come to the port where the server is waiting for connection requests.this socket will be connected to client for further communication with client and the listening Socket will do its job of listening for communication channel requests.

3. Now a buffer is created for each communication socket and the serverwiill be checking for the condition for the buffer not empty . If it is true server will read the data into the message datastructure and According to Command it will do different jobs as mentioned below

    1. Send the list of Boards available in the server to client

1. open the file Boardlist.txt add the file names in the file to the board list in the message structure and set command sccordingly in the message structure
2. Send message to client

2. Send the particular Board requested by the client
1.open the file Boardname.Brd Load it to the Message Datastructure
2. set the access rights flag
3. send message datastructure to client.

3. Save the modified Board sent by the client.
1. Load the board from boardname.brd file some temparory Board Object;
2. recognise the modified pages then replace the modified pages
3. recognise the new pages and add to the existing board

4. Once the modified baord is saved send the acknowledgement to the client about the modified board received.

5. Save the Board created by the client.
1. open the new file Boardname.brd store it in that file
2. add board name to Boardlist in th server

6. Send acknowledgement to the client after the board is saved on this machine

7. Delete theboard after checking the passwordsent by the client to delete the board created by that client.
1. open the file Boardname.brd Load the contents to the temparory Board object
2. check password in the message datastructyre with the Temporary Board Objects password if matches delete the Boardname in the Board list and delete the file Boardname.brd

Let me talk about something about the storage of the Boards and list of Boards available on the server.

Each Board will be stored in a file named "Boardname.Brd" .

List of Boards will be stored in afile Named "BoardList.txt "

Both these files will be created by the server in a directory where it is executing the program.

4.3 Client Algorithm.

1. Ask for port and network address of the server and Create a Socket search for the machine in the network and for the port on the machine if sever is listening on that port then connection will be established and a window will be displayed on the terminal. If any error occurs in the proccess then giv eerror message and ask for retry and change of address .

2. Once the connection established itcan send message for

    1. Board is created save it in the server
    2. List of boards in the available in the server

        1. send a particular board selected in the list box.
        2. Board modified save changes.
        3. Delete a particular board if the password given by the user matcheswith the Bord password.

3. it will check for the condition buffer is not empty if it is true it will read the message into the message datastructure and checks for the command according to that server will do the following things

    1. Boardlist have come Store it in the Boardlist Object of Document class in the client.
    2. Board has come Add it to the Boards available List in the document.
    3. System Message has come display it in the window.

Conclusions

We have implemented the above said algorithms for PBBServer and Client on Windows95 Operating System using Microsoft DeveloperStudio and used MFC classes. We tested the program with different data with all possible access rights given by the creators of the board.

Guidence to the user

By this time you might have realised that programming of this application is done in VC++.If You have source code with you then open these projects in project work space of microSoft developerstudio Build the projects . Now run the server.exe on your machine or on any other in the network, on dos prompt or in the MicroSoft Developer studio. It will ask you for server port enter a number in it and then press OK if you can see awindow on your terminal then the server is running on the specified port else change the port number.

Now start the client on the dos prompt or on the microsoft DeveleporStudio on any machine in the net work it will ask for address and port of the server give it if the address and port specified is the same as the serevr you will see awindow on the terminal else it will ask for retry for connection and then change of address or port.

To executing any of client or server on Dos prompt your command line should support windows and at the same time it should contain all the DLL libraries if it doesn't then program will give an error message on the terminal . Then load that DLL and then execute the program.

REFERENCES

[1] UNIX Network Programming ----W.Richards Stevens,Prentice Hall Of India Pvt Ltd.,NewDelhi-110001

[2] Internetworking with TCP\IP : Vol 1 -----Dougles E. Comer. Third Edition Prentice Hall Of India Pvt Ltd,New Delhi-110001

[3] Windows Programmimng with MFC ----Peter Norton ,Prentice Hall Of India Pvt Ltd. NewDelhi-110001.