

A SYSTEM OF COLLABORATIVE WRITING AMONG WRITERS THROUGH THE INTERNET

Suraya Masrom & Hafiza Abas

Unit Sains Komputer, Jabatan Sains
Program Pengajian Diploma
Universiti Teknologi Malaysia

ABSTRACT

This paper focuses on collaborative writing developed using client / server computing and the Internet. The client application consists of Multi User Text Editor, which provides basic facilities for users to edit shared Rich Text Format (RTF) document. The RTF document is stored in a server and can be edited and saved by a client at a time. The latest updated file will be sent back to team members if the servers received refresh command from the client.

The server is a program running on a known host, which is responsible for shared file storage, access control and managing concurrent access between different members of the same workgroup. All activities such as log on, log off, open and save document will be logged into a file by the server that team members can view it.

In order to develop connection between client and server, this system uses Java's Internet Protocol Suite provided in java.net package. This package provides a set of classes that support network programming using the communication protocols employed by that Internet includes TCP/IP.

Keywords: Client, Server, Server Socket, Client Socket, Stream.

Introduction

Collaborative writing can be defined as a writing process by more than one writer. In collaborative writing, tasks are discussed, planned and assigned by team members. Documents are often the goal of the collaboration.

Nowadays, the collaborative editing of documents is a very common task. How do we break up the jobs, assemble the material and then give everyone the opportunity to comment on the document before submitting the final masterpiece could be a complex task especially when those writers are not in the same location.

Writing groups are often distributed over many locations because of the globalization of organizations and the increasing interdisciplinary of tasks. Economic and organizational factors are creating workgroups spread over continents. Since many writers already use computers for their jobs, providing computer support for the collaborative writing process has been identified as an important goal.

When considering computer tool support for the writing process one has to take into account the fact that the size and amount of textual documents have drastically increased. This is because tasks nowadays have become more and more complex. The critical point at which some documents become too large for a single person to handle was reached long ago. As a consequence, the creation of these documents is only manageable by teams. The interdisciplinary of tasks has also contributed to that need.

In order to sit together while teammates not in the same location, the system or tool of collaborative writing should be designed in client /server architecture which establish connection among them via the Internet. The word of collaborative requires that the system should support multi-user environment connected to a server and provide services for writers to writing, revising and commenting document like the normal meeting circumstances.

The Concepts of Collaborative Writing

What is collaborative writing? One definition is: activities involved in the production of a document by more than one author, then pre-draft discussion and arguments as well as post-draft analyses and debates are collaborative components (Keen, 1996). Based on this definition, the collaborative writing process includes the writing activity as well as group dynamics. Another definition is "...any piece of writing, published or unpublished, ascribed or anonymous, to which more than one person has contributed, whether or not they grasped a pen, tapped a keyboard, or shuffled a mouse," (Muller, 1997).

There are several degrees of collaboration in authoring. At one end of the range is a single author who through discussion with and review by colleagues produces a document. The other end of the variety is a group of writers who jointly author a document. Research on writing groups (Jankowski, 1997) has identified three coordination strategies for group writing: parallel, sequential and reciprocal. With parallel coordination, the writing task is divided into sub-task, which are assigned to each group member. These tasks could be complete concurrently. Sequential coordination involves dividing the writing task in such a way that the first part of the task must be completed before any other portions of the task. With the reciprocal strategy, the group members work together; simultaneously on the writing task. This is what we highlighted in this project.

How the work is divided would be an issue. Based on the results of the study conducted by Ede and Lunsford(1993), there are six organizational patterns for collaborative authoring were identified. These patterns are:

- The team plans and outlines the task, then each writer prepares his/her part and the group compiles the individual parts, and revises the whole document as needed.
- The team plans and outlines the writing task, then one member prepares a draft, the team edits and revises the draft.
- One member of the team plans and writes a draft, the group revises the draft.
- One person plans and writes the draft, then one or more members revises the draft without consulting the original authors.
- The group plans and writes the draft, and then one or members revise the draft without consulting the original authors.
- One person assigns the task, each member completes the individual task, one person compiles and revises the document.

Nevertheless, all patterns above could bring to many problems for co-authors working at different locations. According to Jankowski(1997), Nowadays, distributed teams often use the following scheme to produce a document together:

- Work begins with a division of work and responsibilities.
- Then the co-authors produce their parts of the document separately.
- Finally the parts are distributed for annotating and for assembling into the final document.
- Communication between co-authors is done by telephone, fax or e-mail.
- During each stage, everyone in the group has to get together through a meeting especially at the last stage reviewing the final draft after assembling, the document.

Although communication between authors can be done through telephone, fax or e-mail but everyone in-group could not be together as the normal meeting. The most important is, they cannot review and edit document at the same time and at different location.

System Architecture

Figure 1 shows the system architecture.

The Client

The client side runs the **Text Editor Window**. The shared file is displayed in the **Text Editor Window**. For a user to be able to modify the file being displayed, first he must request to open file for write access by sending the request to the server. Requests for write access to a file are granted in a *first-come, first-served* basis. Once

the request is granted, the user can edit the file at will, and all changes to the file are propagated to each of the shared file.

The editor provides the user with a set of basic editing functions, such as **Copy Text**, **Cut Text** and **Paste Text**, text insertion and deletion capabilities, as well as options to **Open** and **Save** a file, and other associated options.

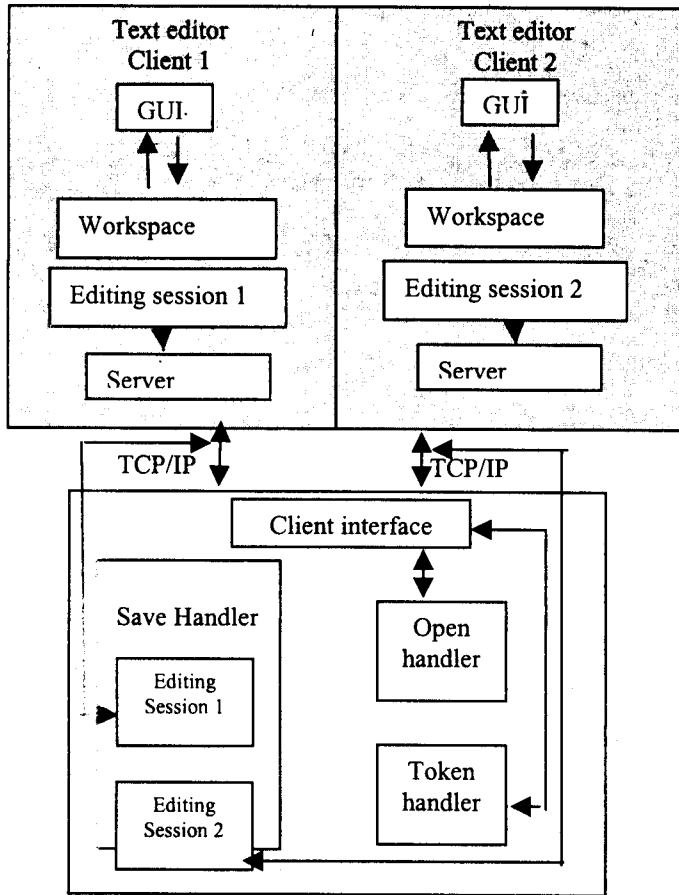


Figure 1: System Architecture

The Server

The server is a program running on a known host, which is responsible for shared file storage, access control and managing concurrent access between different members of the same workgroup. The server act as storage of the entire set of shared files. Each member of the group is capable of accessing these files, but the shared files are not accessible to the rest of the world. It is responsible to handle token passing for each clients. Client who is getting the token can edit document in editor and save the file into server. Server will log every activity of users and record them in a file for subsequent viewing.

Client/Server Communication

Clients and servers establish connections and communicate via *sockets*. Connections are communication links that are created over the Internet using TCP/IP. Firstly, the socket-based server accepts new socket connections and then creates objects that interact with the real application and pass the results back to the socket-based client. The socket-based server itself never interacts with the application server. Figure 2 illustrates a socket-based client connecting to the socket server.



Figure 2: Client/server connection

Secondly, the server creates a socket-based client object, which implements the application's client interface. The socket server also tells the new client object where to find the application object. Figure 3 illustrates this step.

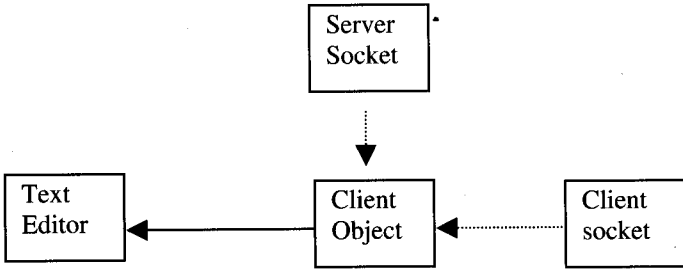


Figure 3: Client Object and application

Finally, the socket-based client object interacts with the application, passing information over the socket connection to the user. Figure 4 shows this interaction.

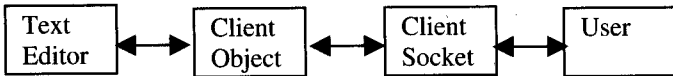


Figure 4: information flow between user and application

TCP socket and Java

Java provides good support for TCP sockets for both client and server, in the form of two socket classes, *java.net.Socket* and *java.net.ServerSocket*. When writing client software that connect to an existing service, the *Socket* class can be used. When writing server software to provide a service, the *ServerSocket* class can be employed.

Client Sockets

To communicate with a remote host using TCP/IP, the client must first, create a *Socket* to the remote host. This automatically establishes a TCP connection, throwing an exception if it fails. In addition to specifying a host name, it is necessary to specify a TCP port, an integer between 1 and 65535.

The easiest way to create a socket is to specify the hostname of the machine and the port of the service. For example, to connect to a server on port 1234, the following code might be used:

```
try
{
    // Connect to the specified host and port
    Socket mySocket = new Socket ( "172.161.32.4", 1234);
    // .....
}
catch (Exception e)
{
```

```

System.err.println ("Err - " + e);
}

```

Once a *Socket* has been created, the *Socket* subroutine *getInputStream()* and *getOutputStream()* can be used to obtain streams through which the client can communicate with the server.

Server Sockets

The *ServerSocket* class is the mechanism by which a server can accept connections from clients across a network. The basic procedure for implementing a server is to open a *ServerSocket* on a particular local port number, and then to wait for connections. When a socket is created from the subroutine *accept()*, a connection has been successfully opened between the client and the server. Clients will connect to this port, and a connection will be established.

If everything goes smoothly (connection establish), information will be sent and received between the client and the server. The two methods in the *Socket* class responsible for that are *getInputStream()* and the *getOutputStream()*.

Stream

Streams are the underlying abstraction behind communications in Java. A stream represents an endpoint of a one-way communications channel.

The communication channel usually connect an *OutputStream* to a corresponding *InputStream*. Everything that is written to the *OutputStream* can subsequently be read from the *InputStream*. This connection can take the form of a link through a network, a memory buffer between threads, a file, or a terminal to the user. Streams provide a uniform data interface for applications, no matter what communications channel is actually used.

Hardware and Software

Minimum hardware and software requirements to develop and run the application:

Client

Client comprises of:

- Computer with Pentium or equivalent microprocessor.
- Computer Memory - 32 MB.
- Java Development Kit (JDK) version 1.3 (for development only).
- Java Station (for development only).
- Java Runtime Environment version 1.3 (for running the application only).
- Internet capabilities (network card / modem).

Server

Server requirements are as follows:

- Computer with Pentium or equivalent microprocessor.
- Computer Memory - 64 MB.
- Java Development Kit (JDK) version 1.3 (for development only).
- Java Station (for development only).

- Java Runtime Environment version 1.3 (for running the application only).
- Internet capabilities (network card / modem).

Advantages of using Java

There are many reasons why Java is used to develop this system:

Meta-Platform

Java has defined for itself a platform that is completely independent from hardware implementation and provides cross-platform API. Considering that the Internet is ubiquitous, it is a relief not to have to worry about writing code that can run on all the possible variations of hardware that exist on the Internet.

Dynamic Interpreter

When we compile a Java application, there is no linking phase as with other languages. The Java virtual machine performs this linking stage at runtime on an as-needed basis. Because all references to classes are loaded at runtime, we can easily replace a class file without replacing the entire application.

Client / Server Language

The *java.net* package provides several classes that support socket-based client/server communication. In fact, Java provides networking protocols employed by the Internet.

Java is free

As opposed to Visual C++ or any other programming language, Java is totally cost free. This alone is a big advantage for any developer who has limited resources.

GUI in Java is free

In addition to the language itself, Sun Microsystems, the creator of Java also provide the graphical user interface facility (API) for cost free. In the early days one would have to pay a large sum of money to use the object oriented code written by others.

Advantages of the collaborative writing system

The system consists of two parts: the client and the server application. The client application is a text editor as well as a network application.

Many implementation of a text editor in Java only support Plain Text, while this project enable the use of Rich Text Format (RTF). This is an advantage considering all the formatting of the document can be done as the writer(s) is building the document right from the beginning.

In case of an editor that only support Plain Text, user will create their document and then will have to use another Word Processor to put some formatting into their writing. This feature alone is a time saver and not to mention the RTF format is a very well known and supported by a lot of commercial applications.

Another advantage of the system emerges from the choice of the development language. Java is a platform independent, which means the code that is written on Windows 98 environment can be recompiled and executed on other environment such as Unix or Linux or even Macintosh.

The uses of Java-Swing package in the design of the Graphical User Interface enable the application users to be familiar with one standard interface. Users will not feel awkward if they are required to use the same application on other operating system. This is because buttons and scroll bars will look exactly the same regardless of the

operating system in use. This feature although seems a minor advantage is not least important because it has a psychological effect to users who are already familiar with a “*look and feel*” application.

Limitations of the collaborative writing system

The system has few limitations:

- At any time only one user is given the right to edit a document. Users will have to take turn if they want to modify the document. To overcome this limitation, however, users can work on an offline copy of the document and later upload it into the server.
- The password system is very weak. Hackers can take advantage of the system very easily.
- The number of fonts is very limited. Further work should be done to increase the number of font and other formatting features of the editor.

Improving the collaborative system functionality

Further work can be done to improve the system such as adding a password system.

The system should provide awareness mechanisms among online writers. For example, in the main text window, color-coded selection, telepointers and shared scrollbars provide awareness of the location and actions of others directly in the focus of the user’s attention. The shared scrollbars can show user locations and allow the local user to synchronize the view to that of a given remote user, for optional WYSIWIS(what you see is what I see) view sharing.

The system should provide “live” annotation mechanism, where users can add comments to the text, suggest alternate wording, include rationale for changer or leave messages for other users. The annotations are “live”, meaning that annotations are immediately available to all participants in the session, can be viewed and edited by all group members.

The potential of collaborative writing system through Internet

Today, the primary use of computer is for document processing. According to Doner(1993), most people who write large texts use computers to support the task of document processing. In a survey, Doner discovered that in the year 1992, 74% of a large number of professional writers already used computers for writing and 11% were considering buying one.

When considering computer tool support for the writing process, one has to take into account the fact that the size and amount of textual documents have drastically increases. This is because tasks nowadays have become more complex. The critical point at which some documents become too large for a single person to handle was reached long ago. As a consequence, the creation of these documents can only manageable by teams. The interdisciplinary of tasks has also contributed to that need. Because of the globalization of organizations, the members of the teams are often distributed over many locations. Economic and organizational factors are creating workgroups spread over continents.

Internet is the best medium that we have today to enable co-workers to work together regardless of the geographical boundaries. There are several aspects however to consider when we chose to use Internet as a medium. The expansion of the cyberspace reduces the transfer time of data. Application should be designed in consideration of this phenomenon. One of the solution could be the use of data compression technology.

Another aspect that has become very important is security. A medium that is accessed by millions of users is in itself a threat to whatever system that is put online. Everyday hackers infiltrate many websites creating havoc and losses to the online companies. Thus, developers should always think of security as an important aspect in the design of their application.

Conclusions

Collaborative writing is a social process because it creates a sense of community, where members construct idea and comment through dialog. Active participation plays an important role in collaborative writing. Therefore, collaboration reduces the gap between writers.

There are many ways how writers can manage the collaborative writing process. Writers can sit together in a meeting to define task, discuss and contribute ideas to team members. Otherwise, they can pass up document to

each writer and communicate via e-mail. The first routine would giving problem to those who are not in the same location while the e-mail system is not really present a meeting environment where team members can talk together at the same time. As a consequence, the best approach is by using an on-line collaborative system where writers can review, write, comment and discuss via the Internet, like normal meeting.

REFERENCES

- [1] Dillon,1993, *Design Issues and Model for a Distributed Multi-User Editor*, working paper MIS 93/10, Graduate School of Business, University College Dublin.
- [2] Doner,1993, *The Use of Computer-and Internet*, International Journal of Internet, Vol.7, pp.293-303,1993.
- [3] Ede and Lunsford,1993. *What Does Collaborative Means?* , working paper MIS 93/10, Graduate School of Business, University College Dublin.
- [4] Ford, Weber, Axxouka, Dietzler, Streeter, and Williams,1999, *Borland Jbuilder 3*, Indiana, SAMS.
- [5] Lemay, et.al,1996, *Teach Yourself JAVA in 21 Days, Professional reference Edition*, Indiana,SAMS.
- [6] Mark Wutka, et. al, 1997 *Java Expert Solution*, Indiana, QUE.

References From Internet

- [1] Jankowski, 1997, *Collaborative Writing Strategy*, http://www.tapr.org/~ird/Collab_Writing/collab_strategy.htm.
- [2] Keen , 1996, *Collaborative Editor*, <http://sern.ucalgary.ca/courses/keen/611/f97/grp2/report.htm>.
- [3] Muller, 1997, *Design Issues and Model for a Distributed Multi-User Editor*, <http://www.informatik.tumuenchen/publication/Koch1997b/paper1.html>