

# MessagePlus: Tool Support for Communication and Collaboration via Internal UNIX Networks

Xiaoran ZHENG, Haiyang SUN, Ricky LEW, Sergiu DASCALU  
Department of Computer Science, University of Nevada, Reno  
Reno, NV 89557, USA

and

Cristian BIRLONCEA  
IBM Toronto Software Laboratory, 8200 Warden Avenue  
Markham, ON L6G 1C7, Canada

## ABSTRACT

This paper focuses on the UML-based specification and design of MessagePlus, a novel instant messenger developed as a tool for communication and collaboration for users working on an internal UNIX network. An overview is first presented to introduce MessagePlus and identify the unique features of this software. The specification phase of MessagePlus is detailed with functional and non-functional requirements as well as use-case modeling in UML notation. The design phase is presented with state and flow charts to highlight the three major components of MessagePlus: text-based communication, graphical communication through a drawing whiteboard, and file transfer. Finally, implementation results are presented via user interface snapshots. The paper concludes with directions of future work and enhancement for MessagePlus.

**Keywords:** Instant Messengers, UNIX Network Tools, Network Communication, UML-Based Specification, UML-Based Design.

## 1. INTRODUCTION

In today's growing technological world there are many means of communication through the Internet [1, 2]. Instant messengers have become very popular as a main option for communication [3, 4, 5]. Instant messengers, however, are not popular amongst educational and business institutions due to the lack of security and hard-to-use functionality that these messengers provide. MessagePlus, a novel software application that we have developed recently, offers a range of facilities that assist students as well as business employees to communicate more effectively within their internal UNIX network infrastructures [6]. In essence, MessagePlus is an enhanced communication tool for internal UNIX networks. The software provides a convenient interface that allows users to communicate within their networks with ease. Users of MessagePlus have the ability to

directly exchange information with other users through *direct text communication, graphical communication via drawing whiteboards, and file transfer*. A user of MessagePlus can engage in a chat with just another user or can join a multiple chat session where multiple users can communicate simultaneously. The whiteboard option provides users a live drawing interface to share with each other in real-time. MessagePlus also provides the ability to send files to one another within a network. The main intent of MessagePlus is to broaden the usage of UNIX machines by providing a friendly interface that is intuitive and easy to use. The most important advantage of this software resides in its support for user communication and collaborative work, including joint study and learning, network-based teaching, brainstorming, and project development.

## 2. SPECIFICATION

Based on an analysis of the needs of the users, the constraints of the UNIX network and existing tools, functional and non-functional requirements are defined to fulfill the development goals of MessagePlus. Use-case modeling in UML notation [7, 8] is also used to further define, in detail, the requirements of MessagePlus.

### 2.1 Functional Requirements

The following is a list of functional requirements for MessagePlus.

*R01* MessagePlus shall allow any UNIX user logged onto the system to send messages to another user logged onto the system.

*R02* MessagePlus shall allow multiple users to communicate within their local area network.

*R03* MessagePlus shall allow users to end a chat session at any time.

R04 MessagePlus shall allow users to maintain a list of collaborators in which they wish to communicate with.

R05 MessagePlus shall allow file transfers to occur between users.

R06 MessagePlus shall allow users to either receive or reject a file that has been requested to be transferred.

R07 MessagePlus shall allow users to use a whiteboard to communicate through simultaneous sketching and editing.

R08 MessagePlus shall allow users to draw and move on the whiteboard predefined lines and shapes such as circles, rectangles, ellipses, dotted lines and dashed lines.

R09 MessagePlus shall allow users to use different colors and pen sizes when drawing on the whiteboard.

R10 MessagePlus shall allow users to save a whiteboard session as an image.

## 2.2 Non-Functional Requirements

Below is a list of non-functional requirements for MessagePlus.

N01 MessagePlus shall be implemented in C++.

N02 MessagePlus shall use QT [9, 10] libraries for the development of GUI components.

N03 MessagePlus shall be easy to compile and install on a UNIX platform.

N04 MessagePlus shall be fast, reliable, secure and user-friendly.

N05 MessagePlus shall be able to run on Linux, a variant of the UNIX platform.

## 2.3 Use-Case Modeling

Each use-case encompasses one or more functional requirements of MessagePlus. The use-case diagram in UML notation [7, 8] is shown in Figure 1.

UC01 The user establishes a connection with another user who is currently logged on to the network. The user enters a message through a MessagePlus interface designated for text communication.

UC02 If MessagePlus is turned on, the user receives a message instantly if another user has sent a message designated for this user. An interface designated for text communication pops up automatically to display the text.

UC03 The user chooses the option to start a multi-user chat session and invites one or more users to join this session. Message exchanges among these users occur through the same type of interface designed for two-user text communication.

UC04 At any time, the user can end a communication channel. The user on the other side of the connection is notified.

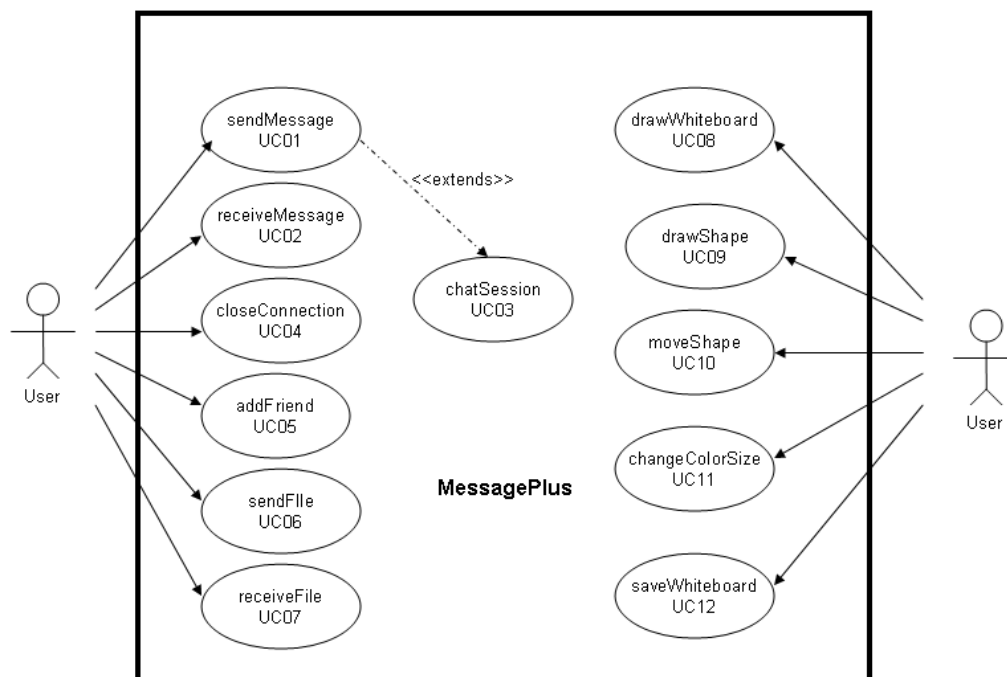


Figure 1: Use case diagram

*UC05* The user can add any user from the network into a personalized list, which facilitates the communication process by allowing the user to simply reference any user that has been added to the list.

*UC06* The user establishes a connection with another user who is currently logged on. The system prompts for the path name of the file to be sent. The user specifies the file and waits for response.

*UC07* The user at the receiving end is notified that a file has been sent to him/her. The sender's username and filename are displayed. The user now has the choice to either accept or reject the file. Regardless of the action taken by the receiver, the sender gets the corresponding acknowledgement. The user accepts the file by an affirmative acknowledgement. The system prompts for the directory that the receiver wants to place the file in. The receiver enters the directory path and the file is placed in the specified location.

*UC08* The user establishes a connection with another user and chooses the option of whiteboard. Users at both ends can draw freehand simultaneously through a drawing interface and see the results in real-time.

*UC09* The user chooses a predefined line or shape such as a dotted line or a circle, then drags the mouse to draw the line or shape. The length of the mouse

movement defines the parameter of a shape being drawn.

*UC10* The user clicks on any predefined shape that has been drawn on the whiteboard, and moves the object to another location on the board. Changes are reflected to the whiteboards of users at both ends.

*UC11* The user chooses any predefined color or size to draw freehand, lines or shapes.

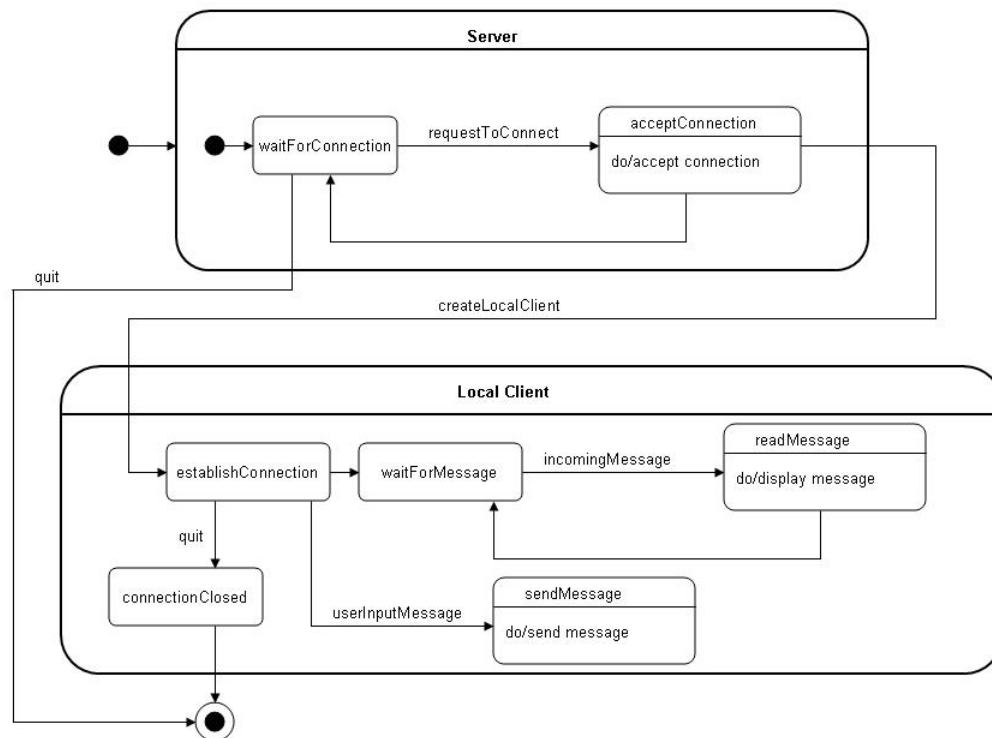
*UC12* The user can save a whiteboard session at any time as an image.

### 3. DESIGN

The design of MessagePlus is composed of three major components: text communication, whiteboard, and file transfer.

#### 3.1 Text Communication

Communication between two nodes within an internal network is established using the TCP/IP protocol suite [11]. At the transport layer of the network encapsulation, TCP is used to transmit packets to a destination specified by an IP address. However, at a higher level of encapsulation where applications need to communicate with each other, a port number has to be used for a specific application to send and receive data [12]. MessagePlus is based on a client-server model where sockets are used to establish TCP connections.



**Figure 2:** Statechart of the process in which a client is connecting with a server

Figure 2 is a state chart in UML notation that depicts the process of a client on one computer connecting to the running server on another computer. The server runs on every machine within the network as a background process, its main job being to wait for remote connections. To establish a connection between two computers, a client specifies the IP address of the computer to be connected, and makes a request to connect with the running server on that computer. The server accepts the connection and spawns a local client to handle data exchange between the two computers. The server then goes back to its initial state of waiting for remote connections from other machines. Once a connection is established between two computers at the application layer, users can send and receive messages through a MessagePlus interface.

Multi-user chat session extends the client-server model to establish connections among several computers. A multi-user session is initiated by one user, and this user invites others to join a chat session. The running server of this user acts as the exchange point for message passing among the connected computers, since this server is connected to the client of every user in the session. For each message received by a client, it relays the message to every remote client except the originator through the server that acts as the exchange point. Figure 3 is a flowchart of a multi-user chat session.

### 3.2 Whiteboard

The whiteboard feature of MessagePlus utilizes the already established TCP connection between two users for graphical communication. The whiteboard supports three types of drawing capabilities: freehand, lines, and shapes with options of predefined color and pen size. MessagePlus tracks user mouse movements when an object is being drawn onto the whiteboard and sends either points or parameters of the object drawn through the TCP connection. The client that handles whiteboard operations are responsible for drawing anything the local user draws as well as any data that are received through the TCP connection. Freehand drawing is captured point by point. Lines are captured by endpoints. Predefined shapes such as ellipses and rectangles are captured by the starting point, width, and height. Color and pen size changes are also tracked to maintain consistency, which means that the two whiteboards at both ends of the connection are exactly the same at all times, and the changes made by either user is reflected instantly.

Each predefined shape being drawn onto the whiteboard, whether the creator is the local user or the remote user, its parameters are kept in a data structure. This allows the user to select any shape and move the shape to another location on the whiteboard. Since every mouse movement is tracked by MessagePlus, this movement is reflected on both ends of the connection. This truly allows the whiteboard of MessagePlus to be a scratch space for simultaneous sketching and editing.

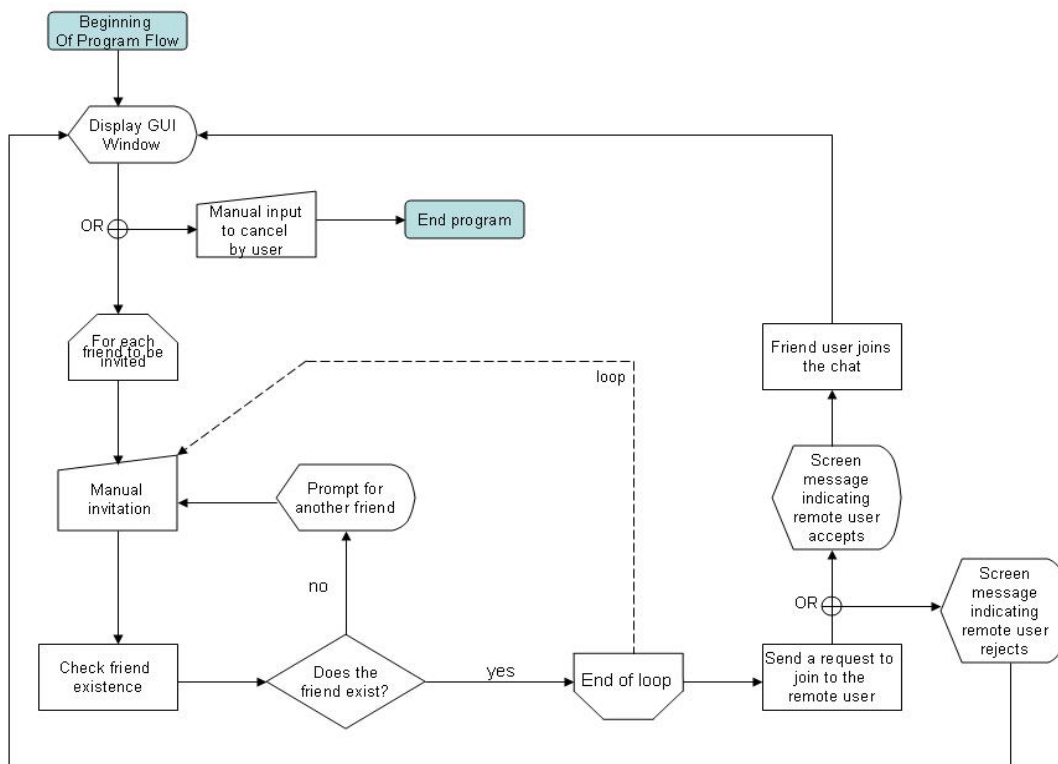


Figure 3: Flowchart for multi-user chat session.

### 3.3 File transfer

MessagePlus' file transfer facility also utilizes the already established TCP connection as a means of data exchange. In the process of sending a file, the user selects a file, MessagePlus checks the existence of the file and transmits a request to send to the remote user. User information and sender information are transmitted over the TCP connection as data that adhere to protocols established to synchronize the transferring process. In the process of receiving the file, the remote user is notified with sender and file information. The remote user has the choice to either accept or reject that file. The response is also encoded as data and is sent over the TCP connection. The sender is notified and MessagePlus acts accordingly to either proceed with or suppress the act of file transferring. There are two types of data under consideration, ASCII and binary. The file transfer capability of MessagePlus utilizes file I/O and socket libraries to send and receive data [11, 12].

## 4. IMPLEMENTATION AND RESULTS

MessagePlus is implemented in C++ using QT libraries [9, 10]. The implementation results of MessagePlus are shown in the following snapshots (Figures 4-7).

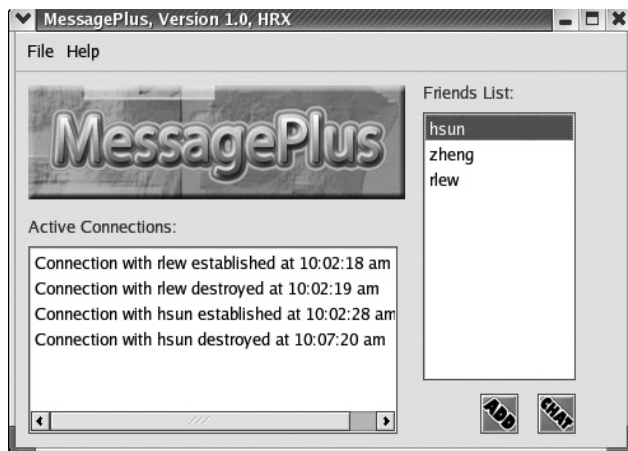


Figure 4: Connection log in MessagePlus

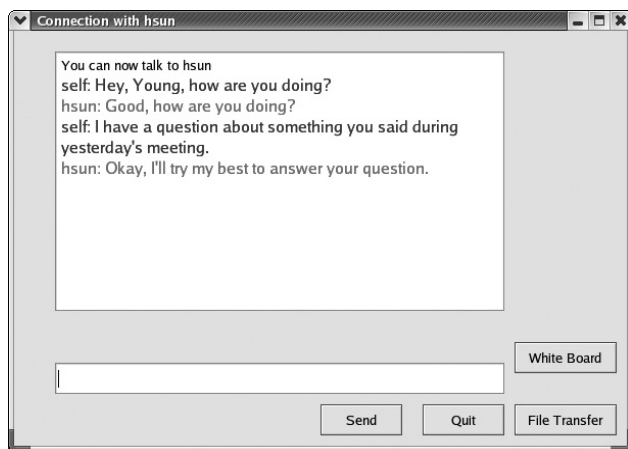


Figure 5: Text-based communication via MessagePlus

A user starts a communication channel by identifying another user whom he or she wishes to communicate with from the collaborators list, as shown in Figure 4, and sends the request through MessagePlus. The receiving user then has a window automatically pop up with the initial message that was typed in by the sending user. Now that a direct connection between the users has been made, they may freely pass messages back and forth to each other until the connection is closed by one of the users, as shown in Figure 5.

A snapshot of file transfer is shown in Figure 6, where a notifier presents the options for the receiver of a file to either accept or reject the file being transferred. The file transfer capability of MessagePlus provides an efficient way to exchange information because of its ease of use and its high speed, the latter due to the high availability of the internal network structure on which the software operates.

The whiteboard feature is shown in Figure 7. The example in the snapshot shows an instructor explaining a calculus problem. As it can be seen, it is much easier to draw an equation and its associated symbols rather than explain the problem using text only.



Figure 6: File transfer via MessagePlus

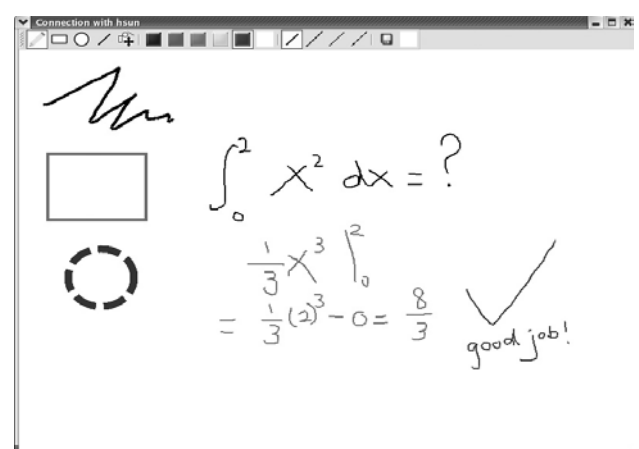


Figure 7: Whiteboard facility in MessagePlus

## 5. FUTURE WORK

One of MessagePlus' planned future developments is the inclusion of encryption to protect the information that is shared across the internal network. The whiteboard feature can be extended by adding sophisticated dynamic operations such as dragging, dropping, and replacing objects. Support for multi-user whiteboard operation is also a desired capability. Another main aspect of future development is voice and video conferencing through the MessagePlus interface, which will make MessagePlus an even more powerful and useful tool for UNIX network communication and collaboration.

## 6. CONCLUSIONS

The main goal of MessagePlus is to broaden the usage of the UNIX operating system by implementing an enhanced graphical communication tool that allows any user to communicate with other users within their network. MessagePlus is a user-friendly program that contains excellent features for communication and networking for all types of users. As such, it is an ideal tool for collaborative learning, communication between instructor and students (or just between students), joint brainstorming, and team project development.

MessagePlus achieves the goal of expanding the UNIX user-base by creating a simple graphical user interface that anyone can use. The software allows users to communicate through their machines either through straight text or a real-time whiteboard for more complicated topics and subjects. Furthermore, MessagePlus gives users the ability to send files within the network with ease rather than using command based programs which can be very complicated for non-UNIX users.

MessagePlus provides new communication features through UNIX networks that have not been implemented before. With all the options provided by MessagePlus, the software is still designed so that it is easy enough for any computer user to operate without external help. In fact, one of the design goals of the tool has been to keep things simple and efficient, thus making it accessible to a large variety of computer users, from beginners to experts.

## 7. REFERENCES

- [ 1 ] Peters, J. D. *Speaking Into the Air: A History of the Idea of Communication*, University of Chicago Press, 2000.
- [ 2 ] Crocker, S.D., "Internet-based Collaboration in 2010," *IEEE Internet Computing*, vol. 4, no. 1, January-February 2000, pp. 53-55.
- [ 3 ] Bird, D., "Choosing an Instant Messaging System," available as of April 22, 2004 at <http://www.instantmessagingplanet.com/enterprise/article.php/2236051>
- [ 4 ] Yahoo Business Messenger 2.0, available as of April 22, 2004 at <http://messenger.yahoo.com/messenger/business/products/msg/sysreq.php>
- [ 5 ] Microsoft Messenger 6.1, available as of April 22, 2004 at <http://messenger.msn.com/>
- [ 6 ] Hall, B., "Beej's Guide to Network Programming," available as of April 22, 2004 at <http://www.ecst.csuchico.edu/~beej/guide/net/>
- [ 7 ] Booch, G., Rumbaugh, J., and Jacobson, I. *The Unified Modeling Language: User Guide*, Addison-Wesley, 1998.
- [ 8 ] OMG's UML Resource Page. Available as of April 22, 2004 at <http://www.omg.org/uml>
- [ 9 ] Trolltech, *QT Whitepaper*, av. as of April 22, 2004 at [www.trolltech.com/products/whitepapers.html](http://www.trolltech.com/products/whitepapers.html)
- [10] Dalheimer, M. K. *Programming With QT*, O'Reilly, 1999.
- [11] Stevens, W. R., Fenner, B., and Rudoff, A. *UNIX Network Programming*, Addison-Wesley, 2004.
- [12] Haviland, K., Gray, D., and Salama, B. *Unix System Programming: a Programmer's Guide to Software Development*, Addison-Wesley, 1999.