DEVELOPMENT OF LAN CHAT MESSENGER (LCM) USING RATIONAL UNIFIED PROCESS (RUP) METHODOLOGY WITH OBJECT ORIENTED PROGRAMMING

Ibrahim Muhammed Abba[1], Mia Torres-Dela Cruz[2], Umapathy Eaganathan[3], Janet Gabriel[4]

Linton University College, MALAYSIA
[1]*starmk29abba@yahoo.com*
[2]*mia@leg.edu.my*
[4]*janet@leg.edu.my*

Vels University, INDIA
[3]*umapathy_mca2k5@yahoo.com*

Abstract

This paper established how LAN Chat Messenger (LCM) using Rational Unified Process (RUP) methodology provides low-cost communication between staff in an organization. In the process, it enables users to make voice calls, chat and transfer of small to medium-sized files. The paper provides information about how voice calls are made, call history is shown, chat conversations are saved and quick file transfers are made through LCM which minimizes the communication issues, which are mainly time and cost of communication and maintenance. Object oriented programming was used to develop the LCM system and the programming language utilized was JAVA, focused on the implementation of two-tier architecture within TCP/IP. The system was implemented based on a standalone application using pure object oriented language. This system was successfully tested at one of the organizations for the domain study.

Keywords: Chat, LAN, Client/Server, LCM, JAVA Programming, Object-Oriented.

1. Introduction

LAN Chat Messenger (LCM) is a chat system via local area network. The functions of the system include voice call, send files, call history, save conversation, chat room, conference calls and so on. According to Dusi, Crotti, Gringoli, and Salgarelli (2008), the bulk of people who used local area networks nowadays put in force security procedures at their limits. The procedures are generally designed to succeed at least two aims. First, they attempt to limit the harm to the network that valid clients may cause, for example, by mistakably downloading virus infected emails or by permeable secret business information to the outdoor world. Second, network edge security strategies can limit or sometimes eradicate difficulties produced by worms and other types of malware.

Object oriented programming (OOP) was used in the development of the system. Sumerville (2007) defines OOP as a software design using object oriented programming language (OOPL) such as C++ and Java. Before 1980, it is a process-oriented programming language which alters the real-life problems into the process, and then accumulates every method to process. With the development of programming technology, to the 20th century, 80's, the object-oriented programming thinking is proposed, which gather a thing (or entity) to procedure and do not transform it into the process, so the probability of errors is bargain (Li and Xu 2010).

The OOPL used is JAVA programming. Java is one of the most popular object-oriented programming languages. According to Yiyu, Chihang and Fong (2006), currently, Java is commonly used in network applications and mobile devices because of its advantages like security, strength and platform independence. First of all, the reason is Java is an object-oriented programming language and the Java computer-generated machine is a stack-based machine, but the conformist computer architecture does not support object-oriented programming in hardware directly. Finally Java is a dynamic binding language, and during execution, many data type and security checking are performed to maintain the system's robustness. Therefore, an exact object-oriented processor targeting Java is a good choice to improve the execution speed.

In this paper, the authors focused much more on some important functions of the LCM such as, voice calls, display of call history, sending files and saving of chat conversations. There are many functions in the LCM which the existing system don't have as compared to the existing LAN messenger in the market with proposed system and it was found that LCM is much better than the existing systems because of some functions such as making voice calls and sending files.

2. Literature Review

2.1 Two-Tier Architecture

The LCM uses two-tier client server architecture as shown in figure 1. The application handling is completed separately for database queries and updates and for business logic processing and user interface presentation. Generally, the networks bind the back-end of an application to the front-end, though all tiers can be present on the same hardware. The architecture of any client/server situation is by classification at least a two-tier system, the client is the first tier and the server is the second.
The two-tier design generally encompasses client demanding services undeviatingly from server i.e. client communicates alongside the server without the help of another server or server process (Chandra & Kumar, 2009).

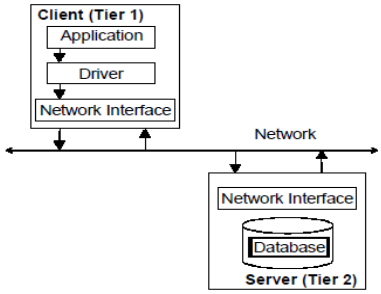

Figure 1: Two Tier Client/Server Architecture Model (Chandra & Kumar, 2009)

2.2 Socket

In order to implement LCM, it needs socket which is an object that represents a low-level access point to the IP stack (Reid, 2004). A socket is a simulated medium that allows sending and receiving of data in an application. The socket enables an application to connect to a network and communicates with other applications connected to that same network (Donahoo & Calvert, 2009). A socket basically constitutes the combination of an IP address and a port number. (Socket = IP address + Port Number).

2.2.1 Types of Sockets

There are many types of sockets that fit into dissimilar protocols and dissimilar stacks of protocols in a suite. However, this paper is focused on the TCP/IP protocol because the TCP/IP is the standard and mostly used protocols in instant messengers. The primary or fundamental sockets in the TC/IP are the stream socket and the datagram socket (Donahoo & Calvert, 2009). Figure 2 below shows sockets and ports.

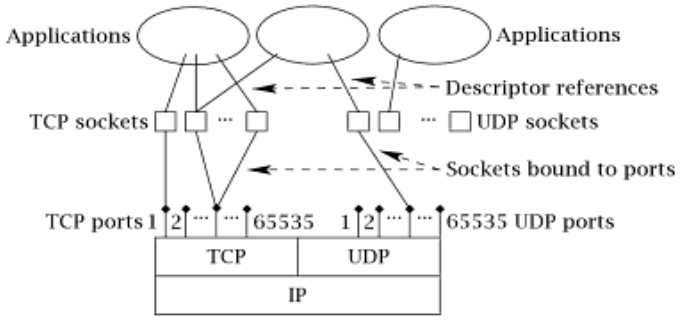

Figure 2: socket and ports (Source: Donahoo & Calvert, 2009).

2.2.2 Chosen Socket

Having known the types of TCP/IP sockets and how they work, in this paper, the authors will employ the stream socket because it uses the TCP protocol which allows limitless messages to be sent over a network unlike the datagram socket that is limited to 65,000 bytes. Besides, it is very secure and more reliable as compared to the datagram socket. Also, it ensures reliable delivery of messages and files.

2.3 How Chat Communications Work

Two sockets are created at the client side and the server side. The client connects to the server through its IP address and port number. They must share the same port number for them to communicate (Reid, 2004).The client and the server both communicate through a stream of bytes written to the socket. The client and the server must agree on a protocol (TCP, UDP or RAW) and agree on the language of the information transferred back and forth through the socket (Umar & Justin, 2003). This study used socket concept to collect message. Basically, the message sent by one staff into a socket and passes it to another staff on the receiving side. If it is group chatting, a central socket will be used to collect the message and then it will be broadcast to all staff that are active.

2.4 Network

Client/server network was used to implement the LCM because of the central administration, scalability- any component or client can be upgraded when required, flexibility – new technology can be integrated into the network should the client increase. Table 1 below shows the comparison between peer to peer network and client/server network.

Table 1: Comparison between the peer to peer network & the Client/Server network architecture

	Peer to Peer Network	Client/Server network
Hardware Cost	It needs no high-end server as the resources are distributed over all clients which reduce cost.	A dedicated computer server (hardware) that distributes resources is needed.
Easy Setup	It is easy to setup mainly if the computers are less than fifty (50).	It is difficult to setup.
Network Operating system	There is no required network operating.	Network operating system is required.
Failure	It can accommodate failure i.e. if one or more Computers (clients) fail the others can still be up.	It cannot accommodate failure if the server fails.
Security	It has security deficiency as clients' administration is not guaranteed.	Very secure because server administration is guaranteed.
Performance	It performs less	Performs very good
Backup	It has decentralized backup that is difficult to access.	It has a centralized data backup with ease of access.

Table 2 below shows the comparison between the existing systems and the proposed system.

Table 2: Comparison between Softros LAN Messenger, Outlook LAN Messenger and Proposed system

Factor	Softros LAN Messenger	Winpopup LAN Messenger	Proposed System (LCM)
Security	It is a secure LAN messenger	It is a secure LAN messenger	It is a secure LAN messenger
Encryption	Yes	Yes	No
Network Connection Type	It requires only the LAN to communicate with other users.	It requires only the LAN to communicate with other users.	It requires only the LAN to communicate with other users.
Protocol	It uses the single protocol, i.e. it can only communicate with other users on the same client software using the TCP/IP protocol.	It uses the single protocol i.e. it can only communicate with other users on the same client software using the TCP/IP protocol.	It uses the single protocol i.e. it can only communicate with other users on the same client software using the TCP/IP protocol.
Communication Feature	No Voice Call	No Voice Call	Support Voice Call
	Slow file transfer	Slow file transfer	Fast file transfer
	No save conversation option	No save conversation option	There is an option to save conversation

3. Methodology

Rational Unified Process (RUP) was used as the methodology of the proposed system. RUP is the process of building a working system by using object oriented concepts to describe work flows and phases of an information system (Bruegge and Dutoit, 2000). The RUP is also known as a procedure instituted and conserved by Rational Software (Manalil, 2011) and is a process-oriented software development model. Also it has a determined system analysis design, implementation and testing (Ge, 2010). RUP is a process context that includes characteristics of designing and configuration for a wide range of projects (Mamaghani, Mousavi, Hakamizadeh and Sadeghi, 2011). However, Sumerville (2007), stated that RUP is not a suitable procedure for all kinds of progress but it does embody a new creation of generic processes. The most vital changes are the separation of periods and workflows, and the credit that employing multimedia in a user's nature is portion of the process. It establishes a methodical method for allocating tasks and obligations inside an enhanced organization. Its target is to make sure that the conception of high-quality multimedia meets the needs of its end-users inside an anticipated design and budget. Figure 3 shows RUP phases and describes work flows of RUP phases.

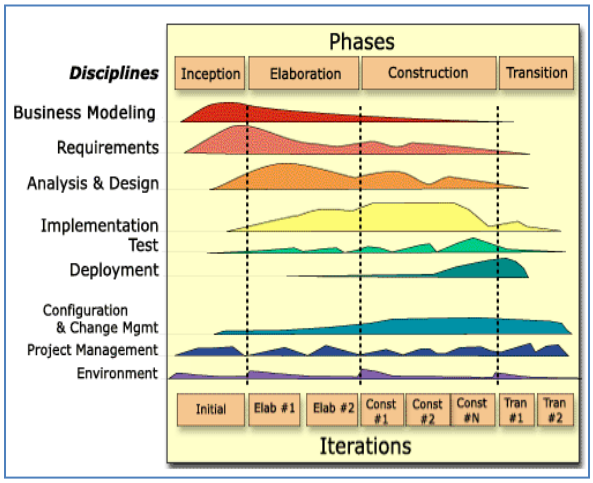

Figure 3: RUP phases (Manalili, 2011).

3.1 RUP Phases

3.1.1 Inception

This phase is the fundamental process of understanding why an information system should be built. It also enables the authors to determine how the project will go about building the system. During project initiation, the business value to the organization is identified. At this phase, scope was also defined and its business case.

3.1.2 Elaboration

Analysis of project needs is done in greater detail during this phase of the software development. This phase will enable the authors to define what the functional and non-functional requirements of the system will be. Also, this phase will help the authors define the aim, scope, feasibility and plan. At the end of this phase the authors will review the requirements to see if these are complete. If it does not, authors will make the necessary corrections, else, they will proceed to the next phase.

3.1.3 Construction

Software design is done at this phase and source code is produced. In this phases the authors will focus on designing the high-level design, the database, the interface and the detailed design which make up the system and observe the rules governing design principles i.e. usability principles. The high-level design will describe the structure of the data and the algorithm required to implement a module of the system. The database design will give a detailed description of the data schema required to support the high-level design. While the interface design will describe how the human computer interaction of the system will be. At the end, the authors will review the design. If flaws are observed they will make the necessary corrections.

At the completion of the design phase, the implementation phase begins. The authors will transform the high-level design into codes and integrate the database design into the codes. After completing the phase, the author will review it and proceed to the next phase.

Also in this phase the authors will separately test each of the modules to find and fix any bug using outlined white box and black box test criteria. Thereafter author will integrate the tested modules and components together. After this, he will then test the entire system again to find and fix bugs.

3.1.4 Transition

Final system is tested and put into operation at this phase. After the system has been installed and in use, environment definitely changes, which will need the system to change in accordance to the environment change. So in this phase, when such changes like new technologies, new methods, etc. occur, the author or other developers that were not involved in the development of the system will have to upgrade the system to meet with the changes of the environment using the proper documentations of the system.

4. Use-Case Diagram

Figure 4 is the use-case diagram of the LCM which shows all functions of the system.

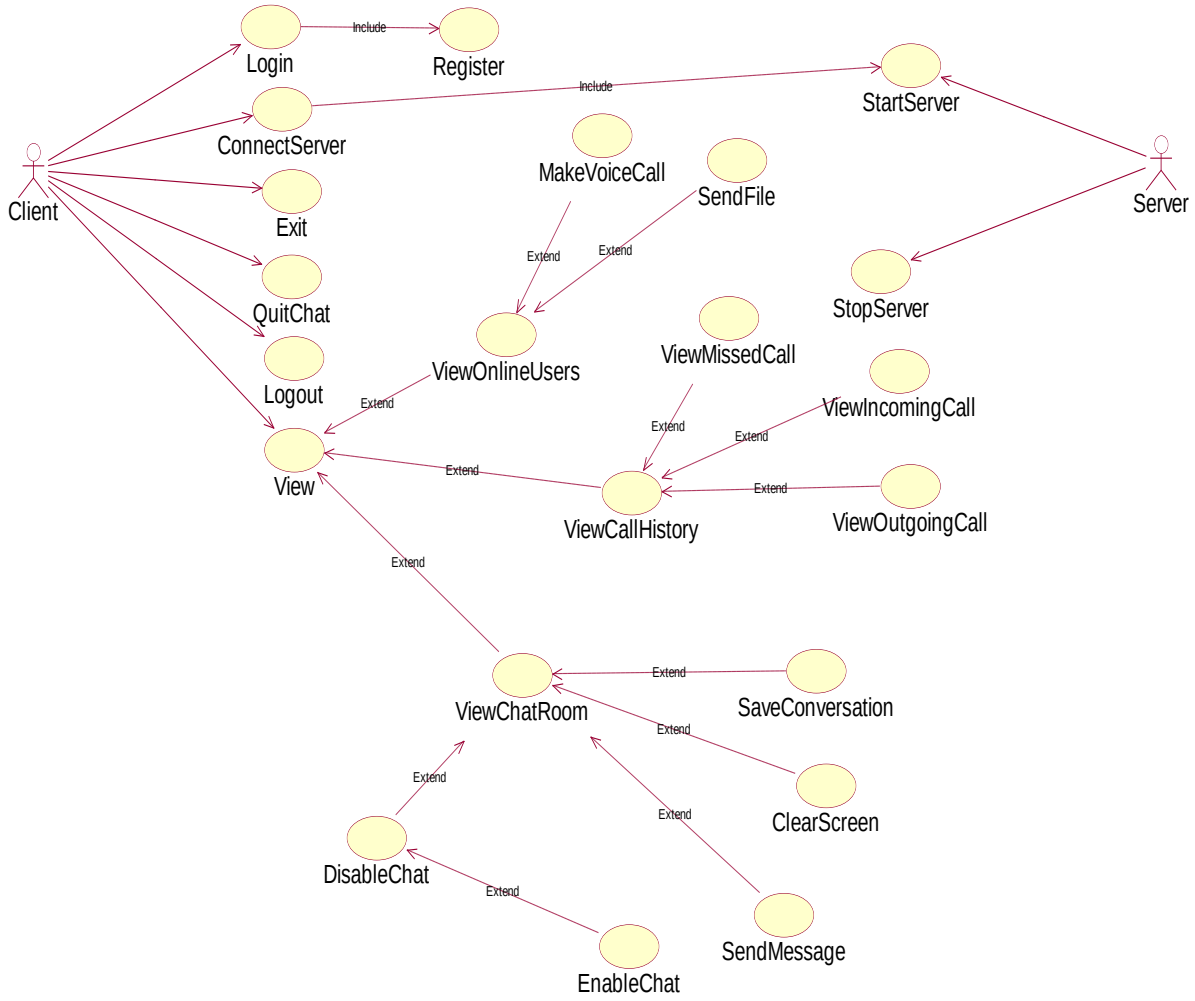

Figure 4: Use-Case Diagram of LCM

5. System Implementation

The system was implemented using Netbeans IDE, using JAVA programming. Below are some interfaces of the system implementation:

The Figure 5 and 6 below shows the voice call process and send file process.

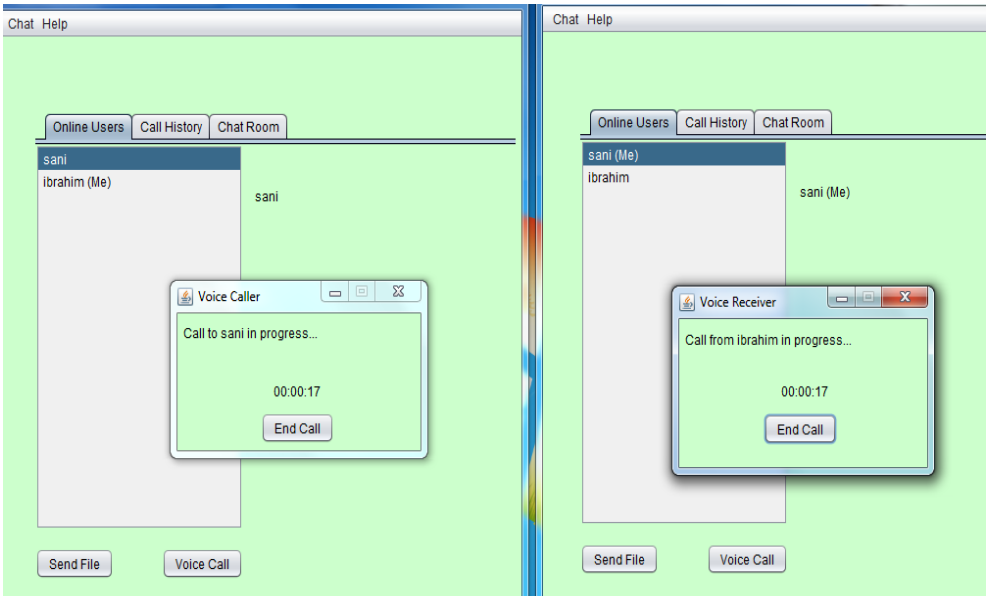

Figure 5: Voice Call Process

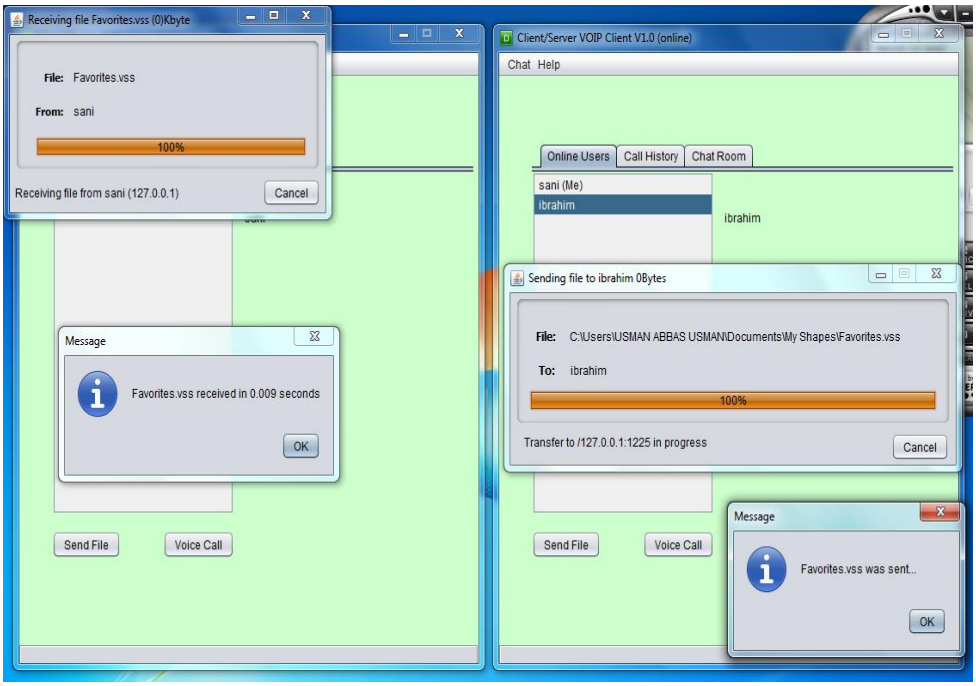

Figure 6: Send File Process

6. Conclusions and Discussion

The LCM is established as a solution to some communication problems and reduces the use of resources including time factor of an organization's internal communication. The proposed system enables users to communicate on networks outside internet boundaries. LCM can be integrated in other application areas such as in school, university and public libraries for communication between library patrons and librarian, in business offices, scientific organizations, and the academe, to mention a few. The proposed system can be a future replacement for many internet chat applications and will cost the organization lesser resources to implement. The strengths of the system identified by this paper are: more communication opportunities within an organization without using an internet subscription; it can work on all Windows platforms provided that the client and server are using the same type of operating system; it provides users to make voice call and send files; it enables connected clients to save their chat conversations; it supports peer-to-peer and server-based modes; user interface is common and familiar; and users may create chat rooms for specific topics or users so if there is any urgent management announcement is easy to broadcast to everyone at once and to specific persons only. There are other advantages of the system especially when future enhancements would be analyzed and implemented.

References

Bruegge, B and Dutoit, A.H (2000). *Object-Oriented Software Engineering Conquering Complex And Changing Systems*, Prentice Hall, Upper Saddle River, New Jersey

Chandra, S.Y and Kumar, S.S (2009). *An Introduction to Client/Server Computing*, New Age International Publishers, New Delhi.

Donahoo, J. M. & Calvert, L. K. (2009). *TCP/IP Sockets in C: practical guide for programmers*, 2nd ed., USA: Elsevier Inc.

Dusi, M., Crotti M., Gringoli F., Salgarelli, L. (2008). *Detection of Encrypted Tunnels across Network Boundaries*. *In ICC*. Brescia, Italy: IEEE.

Ge, C. (2010). *Modifying RUP (Rational Unified Process) to comply with CMM (Capability Maturity Model) levels 2&3*." In ICISE. Hangzhou, China: IEEE.

Li, L., Xu, Y. (2010). *The Teaching Research on a Case of Object-Oriented Programming. In ICCSE*. Hefei, China: IEEE.

Manalil, J. (2011). *Rational Unified Process, Cochin University of Science and Technology.*

Mamaghani, N.D., Mousavi, F., Hakamizadeh, F., Sadeghi, M. (2011). *Proposed Combine Framework of SOA and RUP*. In ICIS. Chengdu, China: IEEE.

Reid, F. (2004). *Network programming in .net with c# and visual basic*, Elsevier digital press, 200 wheeler road, Burlington, MA 01803, USA.

Sommerville, I. (2007). *Software Engineering: Software Process*, 8thedn, Addison Wesley,London.

Umar, S. and Justin, M.P. (2003). *Service-oriented Network Socket*, Association for Computing Machinery Publishers, New York.

Yiyu, T., Chihang, Y., Fong, A. (2006). *Architectural Support on Object-Oriented Programming in a JAVA Processor*. In International Conference on Digital Object Identifier. Hong Kong: IEEE.