Chat Decision Assistant: An Online Distance Collaborative Decision Tool

by

Shastri Stefan J. Sandy

Submitted to the Department of Electrical Engineering and Computer Science
in Partial Fulfillment of the Requirements for the Degrees of
Bachelor of Science in Electrical [Computer] Science and Engineering
and Master of Engineering in Electrical Engineering and Computer Science

at the Massachusetts Institute of Technology
August 22, 2002

© Copyright 2002 Shastri Stefan J. Sandy. All rights reserved.

The author hereby grants to M.I.T. permission to reproduce and
distribute publicly paper and electronic copies of this thesis
and to grant others the right to do so.

Author

Department of Electrical Engineering and Computer Science
August 22, 2002

Certified by

Dr. David Cavallo
Thesis Supervisor

Accepted by

Arthur C. Smith
Chairman, Department Committee on Graduate Theses
Chat Decision Assistant: An Online Distance Collaborative Tool

by

Shastri Stefan J. Sandy

Submitted to the Department of Electrical Engineering and Computer Science
in Partial Fulfillment of the Requirements for the Degree of
Bachelor of Science in Computer [Electrical] Science and Engineering
and Master of Engineering in Electrical Engineering and Computer Science

August 22, 2002

ABSTRACT

Online Chat programs have been proven to be a convenient and efficient tool for synchronous, non-collated communication between both friends and strangers. The ease with which chat discussions allow ideas to be presented unhindered makes it a potentially powerful distance collaborative tool. At the same time, the free flowing communication generates a large volume of noise that makes it difficult to use standard chat logs to quickly understand the concepts generated from a given chat session or over a period of chat sessions. This thesis looks at a program designed to archive chat sessions using concept maps. The maps can either be for stand alone chat sessions or be used to show the progression of ideas over a period of chat discussions. The program is designed to allow groups to arrive at effective decisions based upon their chat discussions.

Thesis Supervisor: David Cavallo
Title: Assistant Professor of Media Arts and Sciences, MIT Media Laboratory
# TABLE OF CONTENTS

1. **Introduction** .............................................................................. 8  
   1.1 Motivation .............................................................................. 10  
   1.2 Organization of Thesis .......................................................... 11  

2. **Goals** ........................................................................................ 13  
   2.1 Problems with contemporary chat programs ......................... 13  
   2.2 Why concept maps .................................................................. 17  
   2.3 Objectives ............................................................................. 19  
   2.4 Example of use of the Chat Decision Assistant ....................... 21  

3. **Previous Research** ....................................................................... 23  
   3.1 Conventional Chat Programs .................................................... 23  
   3.2 Unconventional Chat Programs ............................................... 26  
   3.3 Coterie: A visualization of the Conversational Dynamics within IRC ....... 27  
   3.4 Conversation Trees and Threaded Chat ................................... 28  
   3.5 Aria: An Agent for Annotation and Retrieving Images .............. 29  
   3.6 Babble ................................................................................. 30  
   3.7 Overall Thoughts ................................................................... 31  

4. **Implementation** ........................................................................... 32  
   4.1 Chat Decision Assistant System ............................................. 32  
   4.2 Front End ............................................................................... 33  
   4.3 How to use the system ............................................................ 34
4.4 Database ..................................................................................35
4.5 How CDA works ......................................................................37
4.6 Moderator's viewpoint ..........................................................38
4.7 Regular user's viewpoint .........................................................41
4.8 Concept Maps ..........................................................................42
4.9 Windows Manager / Menu Bar Options .................................43
4.10 Concept Map Menu ...............................................................47
4.11 Implementation Details ........................................................47
   4.11.1 Front End GUI ...............................................................48
   4.11.2 Back End Data Processing ..............................................48
   4.11.3 Concept Map .................................................................51

5. Evaluation of Implementation and Effectiveness .........................52
   5.1 The Front End System ........................................................52
   5.2 The Data Processing ..........................................................53
   5.3 The Concept Maps ............................................................54
   5.4 The Analysis Windows ........................................................55
   5.5 Overall Analysis ...............................................................55

6. Future Direction of Research .....................................................57

7. References ................................................................................59
**List of Figures**

3.1 Example of Avatar Chat .................................................................23
4.2 Chat Decision Assistant GUI .............................................................32
4.3 Color assignment for Types ..............................................................33
4.6 Example of a concept map ...............................................................39
4.7 Checking status of other Chat Channels ...........................................40
4.9 File Options ..................................................................................42
4.9 List of Previous Chats ....................................................................42
4.9 SaveText As Option ......................................................................43
4.9 View Chat Transcripts ...................................................................44
4.11.2 Concept Map Menus .................................................................48
Acknowledgements

I will like to thank the voices that spoke to me throughout my thesis. My thesis advisor David, the members of the Future of Learning Group: JK, the Brazilian, the Thai, the Mexican, the Texan, the Italian and Carlos “the UROP”. I will also like to thank my family, Shiva, Sherry, mom, dad; who constantly motivated me by asking what I did each day. Were it not for them, I would not have had so much to write. Other favorable mentions include: Alisa, Manish, Ashish, Piyush, Su, Joe, Viegas, Stefie, Bill, Anant, Martin, Luciano, Chris, Arjun, Davie, Laura, Sammi, Jade, Joyce, Lin, Ravesh, Kevin, Mario, Arthur, tchan@harvad, the Trinis at MIT, and the crazy people I met through my chat research. Without you all, this thesis will never have been …
Chapter 1.

Introduction

In an increasingly global village, people from around the world are finding it very easy to use the Internet to communicate with each other. Chat programs and their derivatives such as Instant Messenger (IM) are a popular electronic medium for carrying out such discussions. According to a research by the virtual community moderator for TechSoup,\(^1\) for the period 2000-2001, 97% of online consumers sent emails, 51% used instant messaging and 27% used chat rooms. Over 160 billion emails are sent every year in the United States and more than 850 million "instant messages" are sent every day.

The ability of these chat and instant messaging programs to enable free flowing conversations, and to be used in both synchronous and asynchronous manners give them the potential to be effective tools for distant, collaborative, decision-making. One reason why these programs have not been embraced as decision-making tools is the fact that since conversations occur so easily in these mediums, it is easy for topics to be lead astray. These random conversations reduce the usefulness of chat archives. Another reason is the lack of resources that contemporary chat programs offer for analyzing the chat contributions. To make effective decisions users need to be aware of all the important concepts that were presented in past discussions.

\(^{1}\) http://www.techsoup.org
Initial computer systems were designed with the concept that users would perform their tasks in isolation\textsuperscript{2}. The World Wide Web, through its ability to connect users around the world resulted in technology such as mobile code (Java), virtual environments and collaborative tools. These collaborative tools were divided into synchronous communications including chat, audio/internet phone, video teleconferencing and asynchronous communication such as e-mail and bulletin boards.

This thesis will look at a new way of moderating real-time (synchronous) chat conversations with the goal of formulating decisions from the conversations. That is, to use a chat program as a distance collaborative tool. In existing chat programs, a moderator refers to a person or program whose function is to either ensure that certain objectionable words do not appear in the conversation or to direct the flow of the conversation by fielding questions and editing the response. In this thesis, the moderator will be used in an attempt to present an alternative method of effectively archiving these chat discussions, while simultaneously allowing for interactive use of these archived discussions in ongoing chat sessions. At the same time, the moderator will try to not control the flow of the conversation. The thesis will examine the features offered in conventional and unconventional chat programs that are suited for distance collaboration; problems associated with the use of chat programs for decision making, and offer adjustments to these chat programs; and describe the implementation of a chat program that will better archive chat conversations and aid in decision making during chat sessions.

\textsuperscript{2} Abrams, Marc \textit{World Wide Web – Beyond the Basics} Prentice Hall, 1998
1.1 Motivation

Since joining the Future of Learning Group\(^3\), (FoL) at the MIT Media Laboratory I have been exposed to various groups at the laboratory whose members are separated by geographical distances but who also have available to them the most technologically advanced tools to facilitate distance collaboration. Inspite of this ease of access to available technology, only for a few meetings are expensive collaboration tools such as video conferencing utilized. For most discussions, the telephone, e-mail and chat programs are the selected medium. Technology that requires little overhead in setting up appears to be the preferred choice for most distance communications.

Whenever communication between members is constrained by geographical distance, time zones, available telecommunication technology and costs, many times, valuable discussions are not recorded, resources are consumed repeating decisions made at prior meetings and bringing members up to speed on what has occurred.

A system that allows for group discussions to be effectively archived, to update people who are not present at the discussion, and to assist while not directing the decision making process will be invaluable to groups such as those at the MIT Media Laboratory and other groups that consist of a large community of users that are separated by geographical distances.

\(^3\) http://learning.media.mit.edu
1.2 Organization of Thesis

The remainder of this thesis is organized into the following 5 chapters:

Chapter 2 outlines the problems with contemporary chat programs from a distance collaborative view point, the problems that the Chat Decision Assistant (CDA) will attempt to solve, an overview of how design of the CDA will solve those problems and an example of a hypothetical use of the CDA.

Chapter 3 discusses related research in the fields of chat as a medium of communication and systems for decision-making based upon synchronous non-collated communication. It explains features of those projects that are appropriate for synchronous non-collated communication and decision-making and what features could be added to improve the decision-making of those projects. Chapter 3 also explains the difference between the goals of the related research and the CDA.

Once the necessary background information and analysis of contemporary work is presented, chapter 4 explains the general design issues behind the Chat Decision Assistant. It also includes the descriptions of the system’s main components, the protocol for archiving the discussion and representing the necessary information. There is a
manual on how to use the system, both from the viewpoint of the moderator and a regular user and an explanation of the function of the key modules in the program.

Chapter 5 evaluates whether the system achieves the claims set forth in the opening chapters of the thesis.

Finally, the thesis ends with Chapter 6, an analysis of future work that can improve the decision making process of the Chat Decision Assistant and other concluding remarks such as other directions the research could be used for but were not considered in this thesis due to time constraints.
Chapter 2

“Making better informed decisions more efficiently”

Goals

Before describing the goals of the Chat Decision Assistant, CDA, this chapter will first describe the problems associated with existing chat programs that this research attempts to solve. It will then describe how the goals of the CDA help solve the stated chat problems and finally, the chapter will conclude with an example scenario of use of the system.

2.1 Problems with contemporary chat programs for Collaborative Decision Making.

There are several types of chat programs, each designed with a specific group of users. As this thesis is geared towards using chat discussions, centered on a project, to formulate decisions about that project, the features of chat that this thesis targets is a narrow set of characteristics. This thesis is geared towards users who are members of a large community who are separated by geographical distance, who are involved in a project, who have frequent discussions about the project, and where not every member is present for all the discussions. In this manner, the thesis tries to incorporate chat and distance collaborative tools.
Chapter 1 mentioned issues that groups such as the members of the MIT Media Laboratory faced in forming decisions. On a more general scale of users, problems around decision making that are relevant to this thesis that the chat users who are targeted by this thesis face include:

1. remembering what was said in previous chat sessions and the general trend of the conversations,
2. remembering the different ideas generated in a single session and the relationship between each idea,
3. determining what contributions each user made to chat discussions,
4. determining the most effective means of bringing members up to speed on what has been said in a chat discussion and
5. determining how to lead the chat discussions towards decision-making without fundamentally altering the manner in which the discussions are carried out.

The above problems can be grouped into two categories: firstly, recalling what has been said in past chat sessions and secondly using the chat contributions to make effective decisions.

I will now look at how existing chat programs deal with user recollection, from the viewpoint of a distance collaborative tool.
In most chat systems, the most common means of discovering what has been previously discussed is to query the chat group or to listen silently on the chat contributions before attempting to become integrated into the discussion. The first case allows the interrogator to be updated with minimum work on his behalf while the discussion stagnates at the expense of the rest of the community. The second scenario prevents the user from making an immediate contribution to the discussion and does not guarantee that the user will be aware of all the important issues that were brought up in past sessions. In many cases the chat archives are never read as most chat archives present the discussion in a linear manner based upon time. As most chat sessions are free flowing, there will be a lot of useless communication surrounding the information that the user is interested in. For most users, the effort required for this far outweighs the process of querying the community or just listening in silently.

Currently, the most effective means of archiving chat discussions in a manner that encourages users to read the archives are threaded formats with discussions archived according to a common theme. However, applications that support this method of archiving tend to have a ‘threaded chat’ front end. The user goes to a message posting area of a web site and reads the messages of interest. If he wishes, the user can then post a reply to one of the original postings or he can create a new topic and post to that. Programs such as WebBoard that are based upon such threaded discussions are not true synchronous chat applications.
For the second main issue that this thesis addresses, making decisions from chat discussions, existing chat applications that are truly realtime programs are not geared towards decision-making. Realtime being the key word as there exist threaded web messaging boards that solve problems ranging from technical service support for a product or medical questions⁴. The product that comes closest to decision making is NetMeeting, a Microsoft product. The product is free with a Windows Operating system and very easy to use. However, the decision-making results mainly from the audio, video and other visual features (e.g. whiteboards) offered by NetMeeting. The chat application is a very basic text based program. The archiving features are limited and while NetMeeting is appropriate for non co-located, synchronous immediate decision-making, its limited archiving capabilities prevents it from being an effective tool for prolonged discussions geared towards decision making over a period of time.

The Chat Decision Assistant (CDA) is designed to solve the above problems. It utilizes concept maps to encapsulate the key ideas from each chat session. Concept maps were chosen as the medium for archiving the chat discussions as they allow the chat discussions to be documented as themes rather than according to the time the ideas were generated. So say there were six discussions about a project from which three key ideas emerged, instead of having six objects, each one representing a chat session, there will be three concept maps.

By providing a diagrammatic snapshot of the chat sessions, the CDA allows members who were not present at the discussions to quickly understand what ideas were generated.

⁴ http://www.nursingnet.org/boards/jobsem4t/
The concept maps consists of nodes and connecting links. Each node contains a chat
contribution, the user who made the contribution and the time since the inception of the
chat that the contribution was made.

2.2 Why Concept Maps

Concept Maps are derived from semantic nets. The use of semantic nets in computer
science can be traced back to Ross Quillian’s 1968 Ph.D. thesis in which he defines a
semantic network as ‘a graph structure in which nodes represent concepts, while the arcs
between these nodes represent relations among concepts’. Quillian’s work centered on
how natural language is understood and how the meanings of words can be captured in a
machine. The semantic networks acted as a model of associative memory. Semantic
networks also represent the organization of one’s ideas in a content domain.

However, Barr and Feigenbaum argued that in a semantic representation, there is no
formal semantics, no agreed upon notion of what a given structure means. The user is
forced to analyze the relationships between the nodes and to evaluate the links created.
This evaluation from one node to the next node is one reason why semantic networks

---

Prentice-Hall.
have commonly resulted in a linear structure. The linear structure comes from considering the relationship between one node to the next node in a linear path.

While semantic networks are commonly referred to as a linear structure for representing information, concept maps are often non-linear. Concept maps allow users to explicitly represent a number of concepts and their interrelationships. They include concepts, usually enclosed in circles or rectangles and relationships between concepts or propositions indicated by a connecting line (the arc) between two concepts (the node). Concept maps are based upon the assimilation theory of meaningful learning in which knowledge is characterized by a propositional network. This view has traditionally resulted in concept maps representing the information in a hierarchical fashion with the most inclusive, most general concepts at the top and the more specific, less general concepts arranged hierarchically below. In their online paper, Zimmaro and Cawley presented a concept map in which concepts are placed in clusters or network patterns in a more horizontal, non-hierarchical structure\(^8\). By allowing the information to be connected in a non-linear fashion, (not a linear path, from one node directly to the next) concept maps are typically represented as extending from the centre outward. Non-linearity allows concepts to be connected in meaningful ways, while allowing multiple paths to a single concept.

---

The non-linearity of the chat discussions and the multiple conversation threads that lead to and away from a concept can be readily illustrated in a concept map. This makes concept maps effective tools for organizing and displaying the information gathered from the chat discussions.

2.3 Objectives:

"Making better informed decisions more efficiently"

The above phrase at the start of the chapter best describes the goal of the Chat Decision Assistant. With the aid of a moderator, it processes the ideas generated from the just concluded chat discussion into a visual representation that is easily and quickly understood by both people who were not present at the conversation and people who participated in the conversation. Another goal is, while aiding in the decision-making, the system should not dictate the manner in which the chat conversations are carried out.

A moderator is needed to pull the important concepts from the chat contributions. Given the time and resources available for the thesis it was felt that having a moderator highlight the chat contributions will result in a more far-reaching thesis than spending time trying to create a program that will use artificial intelligence to determine what contributions contained a key idea and which key ideas are connected under a common theme.
Coterie, described in more detail in Chapter 3, is a program developed within the Media Lab that takes chat archives and separates the entire chat session into separate, distinct conversations where one conversation could be spliced between two other conversations. A test result of the system’s output and the results of 4 individuals who were given the chat archives to parse showed that separating the chat session into separate discussions results in roughly the same parsing for the 4 individuals and the program with the individuals making more reasonable parsing. Thus having only one moderator in charge of determining the threads a submission falls into will not significantly affect the chat archived information that everyone else will view. The algorithm of Coterie is designed to separate, distinct conversations while the CDA needs individual ideas that could belong to more than one conversation. In addition, Coterie’s processing was not always consistent with humans. Consequently it was decided that it would be more efficient to use a human moderator than to attempt to alter the Coterie algorithm.

Another goal of CDA is to create a system that is similar to conventional chat as possible. The user interface will be primarily an area for entering and viewing text data. In order to archive the discussions to form concept maps, the moderator will have additional add-ons. In most chat cases, ones effectiveness is measured by the speed with which one types (‘talks’). The faster one types, the more information (and noise) one is able to contribute to the discussion. Thus, in order for the CDA system to be effective for chatting, it must ensure that the new features it offers does not distract the users from typing. The addition of the moderator aids the decision making process by ‘eliminating’
noise that fast type might have contributed to the discussion. The noise is eliminated in the sense that it does not appear in the archives of the chat discussion.

2.4 Example of use of the Chat Decision Assistant

An ideal example of the Chat Decision Assistant will be as follows:

There is a group of people from various regions of the world chatting. The members of this group are a subset of a larger community who are all working on a common project. Everyone logs in around the same time. There is one moderator, who is pre-determined and who will be present from the start to end of the conversation. The mechanics of chatting will mostly be the same as in IRC (Internet Relay Chats) or web-based chat text messaging. Add-ons to the conventional chat include the ability to correct ones spelling mistakes, a central database that contains the archived discussions; the information within the database is accessible to everyone; and the ability to use data from archived discussions in current chats. During the chat discussion, participants can check to see if other chat discussions are simultaneously occurring.

From a decision-making standpoint, at the end of each chat session, the moderator creates (a) concept map(s) of the chat. These maps can either be new concept maps or additions to existing concept maps. If an existing concept map is modified, the original version of it
is also kept on the server. Thus over time one can use the concept maps to understand how the concepts within chat conversations changed over time.

The concept maps can also be used synchronously with the chat conversations. Participants can choose to view past concept maps, they can import text labels of concepts from these maps into the chat conversations and link the text messages in the chat conversations to saved concept maps so other members of the chat session can better understand what participants are saying. Concept maps can also be linked. By clicking on one node in the map, one can see the sub levels that are actually another concept map.

During the course of the discussion, the moderator tags comments by participants that he deems useful. At the end of the discussion, the moderator returns to these tagged discussions to create the concept maps. The concept maps are created based upon the category the chat contribution falls into and the time from the start of the chat session that the contribution was made.
Chapter 3

Previous Research

"Describing the Internet as the Network of Networks is like calling the Space Shuttle a thing that flies."
(John Lester)

Since the introduction of IRC in 1988, the world of chat has undergone many evolutions.

Programs that retain most of the features of the early 1990’s chat are referred to as conventional type chat while those programs that take advantage of new mediums of communication and new ways of expressing oneself are referred to as unconventional chat. A chat room is defined as being an area on the Internet where two or more people can have a typed conversation in real time. In a chat room, the messages one types are shown instantly to every other member of the room. Within a particular chat room, the discussions are normally focused around a central theme. Just as the Internet has advanced beyond simply being a network of network, the concept of chatting has advanced beyond simply chat room channels.

3.1 Conventional Chat Programs

There are five (5) major conventional types of online chat:

1. Internet Relay Chat (IRC): This was developed by Jarkko Oikarinen to replace Talk. Users enter a specific channel in which a subject or theme is predominant.

\[\text{http://netforbeginners.about.com/library/glossary/bldef-chatroom.htm}\]
Each user runs a "client" program that connects to the IRC network via another program (server). Servers exist to pass messages from user to user over the IRC network.

2. Web based chat: normally, these are Java interface chat programs such as Lucid Chat that run off a browser, preferably a Netscape browser. There even exist a web based version of ICQ (ICQ2GO) and IRC (WWWIRC).

3. Chat Services exclusive to ISPs: Internet Service Providers e.g. AOL provide chat rooms exclusively to subscribers of their service.

4. MUDs: Multiple User Dimension, Multiple User Dungeon, or Multiple User Dialogues are used for real-time, role-playing games. One accesses these sites by telnetting into a remote Internet site. Each user has an avatar that he can use to walk around, chat with other characters, explore dangerous monster-infested areas, solve puzzles, and even create your very own rooms, descriptions and items.  

One interacts with other users in a virtual environment comprised of text or graphical spaces.

E.g. of MUD games:  Maid Marian MMORPG 3D Avatar Chat Window Mode (800x600 initial size)

Or

Maid Marian MMORPG 3D Avatar Chat Full Screen Mode

10 [http://www.mudconnect.com/mudfaq](http://www.mudconnect.com/mudfaq)
Recommended for systems with high-speed 3D graphics card. (ie. NVidia GForce or ATI Radeon)

With increasing skills in graphical design, MUDs have evolved from the esthetically plain but still very complex text world of dungeon and dragons to the programs on the previous page that need graphical cards and use a lot of imagery to convey the ideas of the chat.

5. Instant Messaging (IM): provide instant text messaging services. In its inception in the early 1980’s, IM allowed for real time, person to person communication between users of bulletin boards and the system administrator. IM expanded its client base in 1988 when AOL incorporated a basic form of the program in its buddy list. This eventually evolved into AIM. The main difference between IM and relay chat (IRC) is in the security. With IM one is generally in contact with fewer people than say in IRC and one knows who is listening to the IM messages. With IRC, one does not know everyone within the channel and there could be unlisted users (nics) who could be eavesdropping on the conversation. Finally, with IM one can choose not to respond to a message, while in IRC, one must respond.

In addition to being conventional, the above five are chat programs that most people can easily acquire and for which there exist a large, vibrant community of members and
explicit online assistance. These programs spend a large amount of resources on devising
new means of communicating, to retain the large number of users and to support free
flowing conversations but do not much resources on analyzing the chat logs. Currently,
the best resource for analyzing the chat conversations is a program called mIRCStats\textsuperscript{11}
that describes itself as being, ‘state of the art IRC channel statistics’. The program creates
an HTML page that contains miscellaneous information extracted from the log files. It
creates graphs of the number of users logged in, the number of messages sent over a
period of a day, week or month. It also illustrates the frequency that a word is mentioned
throughout a time period. Decision-making is not a primary goal or selling point in these
chat systems. This may be a conscious decision on the part of the developers of the
system as using the chat programs for productive meetings seems to be in conflict with a
program that allows friends to informally communicate with one another.

3.2 Unconventional Chat Programs

Below are new chat programs that have been developed within the past few years. These
programs have altered some characteristics of the above five conventional chat programs.
The main goal of these programs is to improve the social interactions within chat rooms.
Sub goals include assisting new comers to the chat session to quickly integrate
themselves into the current conversation without having to ask too many questions, to
avoid repetition in conversations, to allow for visual representation of private and public
conversations. Yet, not much emphasis is placed on what happens after the conversation

\textsuperscript{11} http://www.nic.fi/~mauvine1/mircstats
or what are different and new ways of presenting the chat archive so participants can quickly and easily understand what occurred in previous chats.

3.3 Coterie: A visualization of the Conversational Dynamics within IRC.

This program uses a word-based algorithm to analyze chat logs of one chat transcript to determine when a conversation begins, ends or resumes. While it is usefully for writing more effective chat transcripts, it still leaves all the participants with the entire text to skim. The program was developed at the MIT Media Lab within the Sociable Media Group, the same group that designed ChatCircles.

One difference between the Chat Decision Assistant (CDA) and Coterie is the CDA links concepts and not chats. So it is not the case in the CDA that there is a mapping from a chat session to a concept map, rather, it is the case that as an idea is discussed across several chat sessions, the map associated with that concept is expanded. With Coterie the continuance from one chat session to the next is limited.
3.4 Conversation Trees and Threaded Chat

Threaded Chat was created to overcome problems in conventional text chatting such as:

1. lack of links between people and what they say
2. participants not receiving a moment-by-moment information about the reaction of those who are listening to them
3. the process of message production is separate from message delivery.

The main goals of Threaded Chat were not to aid in decision making or aid in the analyzing of chat logs. The creators believed that by using real time threaded chat they would overcome the social problems they perceived in contemporary chat programs. It was a bonus that using synchronous threaded chat was also ideal for effectively archiving the chat conversations.

In the paper, authored by the developers of Threaded Chat and delivered at the ACM 2000 Conference\textsuperscript{12}, it was revealed in the review of the performance of Threaded Chat against regular text chat, Threaded Chat was rated worst that regular text chat in all of the measures except one. While the idea of using threads to archive chat history is a powerful concept, the problem lies in trying to get the user to continually select which conversation he is replying to. Threaded Chat faces the problem that its unconventional front-end chat interface is not user friendly and its powerful concept of threading the chat is not used in analyzing the chat archives.

\textsuperscript{12} Threaded Chat [Proceeding of the ACM 2000 Conference on Computer supported cooperative work December 2000]
In Chat Decision Assistant, the users communicate as they do in regular text chat so they are more likely to use CDA over Threaded Chat. Instead it is the moderator who determines, independent of the other participants, the different ideas that form the threads of the conversation. The moderator transforms the linear with time text conversation to a concept map that illustrates the interlinking ideas of the conversation(s).

3.5 Aria: An Agent for Annotation and Retrieving Images

Aria, developed by Henry Lieberman in the Software Agents group of the MIT Media Laboratory, matches words typed in an e-mail editor to images in an electronic library. There is a ranking system between the words typed and the images. This ranking system improves with increased use.

While not obviously associated with chat, mechanism of linking words to stored data can be expanded, so say during the course of chatting the mention of a particular project will result in all information associated with that project appearing on screen. This will assist in decision making as well as refreshing the memory of the chat participants leading to more effective chat contributions.
3.6 Babble

Babble is a chat-like communication tool that allows its users to engage in synchronous or asynchronous textual conversations, and provides visual feedback regarding who has recently participated in a conversation.

Two features of Babble distinguish it from other chat systems. First, there is a graphical representation, colored dots to represent the chat participants and larger circles envelop the text that the participants contribute. The dots move around the screen to represent which participant is speaking. Secondly, the chat conversations stay on the server permanently. One does not need to have been a member of a chat session to read the archives.

The background research on Babble does not identify how readily users adopted Babble over traditional forms of chatting. As the program presents a new method of interacting in a chat environment, it will be interesting to know if Babble overcame the user-adoption problems that Threaded Chat encountered. A key difference between Babble and CDA is the archiving of the chat discussions. Babble leaves the contributions on the server in a manner similar to most chat logs where the archives are ordered according to time.
3.7 Overall Thoughts

There has been a lot of work on chat programs and their spin-offs. Just as e-mail has replaced letters in the work place, so to will chat have an impact in the workplace. Most of the new research has been geared to new methods of communicating beyond the text messaging approach. Not many new programs have been able to overcome users resistance to adopt a technology that fundamentally alters the way chat communication occurs. One successful approach has been to use graphics to incorporate ideas and convey emotions that would have been difficult to express with only words.

While there is significant work in the front end of chat programs, not much resources has been devoted to analyzing the chat logs. One reason is that there is no demand for it. Some view chat as a ‘virtual conversation’. When they log off the conversation ends as in real life. If a program were to be used to analyze who said which ideas and how frequently someone contributed, it might be analogous to taping an oral conversation and then analyzing that communication.

Another reason for the lack of research is the characteristic of the users. The work community has not adopted chat as a tool for group meetings and thus there is no incentive to create chat programs that support group meetings.
Chapter 4

Implementation

The Chat Decision Assistant (CDA) is a client-server graphical user interface system. There is a client application that each chat user uses to communicate with one another. There is a server application that handles the requests of the client over the network using TCP protocols. Conversation contributions that the moderator deems important are transferred to the concept map and saved onto a mySQL database.

4.1 Chat Decision Assistant System

The display for the chat conversation consists of a frame with 4 main components. The design was intended to be similar to most conventional chat systems. After reading reports by Marc Smith, a co-developer of threaded chat and based on the suggestions of Fernanda Viegas, co-designer of Chat Circles, it was determined that if the primary goal of CDA was to improve the decision making process from archived chat conversations then that goal will be best achieved by having as best as possible a conventional front end chat interface with the decision making process occurring in the background.

A conventional text based messaging front end is a feature that all chat users are comfortable with and thus users will be more willing to use the CDA system. This overcomes one of the reasons why Threaded Chat was not a commercial success; its
unique front-end method of chatting discouraged experienced chat users from adopting
the technology. To aid in decision-making, the chat logs will have to be processed to
extract useful information for the members of the chat community. This data processing
will have to be run in the background so as to not interfere with the ongoing chat
conversations. That is, the data that will be archived and processed will be guided by the
chat conversations that are occurring. At the same time the system will try to ensure that
the chat conversations are not determined by the data processing that is occurring.

4.2 Front End

The four components of the CDA are:

1. The text entry box (bottom portion of the GUI)
2. The window that displays the text entries from the participants (the left GUI section)
3. A display of all the users currently online (the right section of the GUI)
4. A menu bar.

Figure 1: Chat Decision Assistant GUI
4.3 How to Use the System

System users are divided into two categories: moderator and regular users. Each chat conversation has a moderator who is selected before the chat conversation begins. The moderator needs to be present from the start to the end of the chat conversations. During the course of the chat session the moderator selects portions of the conversation that he regards as being relevant to the actual discussion. To aid the moderator a color-coding format is utilized:

<table>
<thead>
<tr>
<th>Type</th>
<th>Color</th>
</tr>
</thead>
<tbody>
<tr>
<td>Brainstorming</td>
<td>Blue</td>
</tr>
<tr>
<td>Problem Identification</td>
<td>Red</td>
</tr>
<tr>
<td>Event Discussion</td>
<td>Green</td>
</tr>
<tr>
<td>Items to be assigned</td>
<td>Pink</td>
</tr>
<tr>
<td>Other Tags</td>
<td>Grey</td>
</tr>
<tr>
<td></td>
<td>Magenta</td>
</tr>
</tbody>
</table>

Table 1: Color assignment for Types

The types that the conversation can be divided into were based upon the suggestions of the other graduate students in the MIT Media Laboratory, FoL group. Tags are sections of the chat conversation that the moderator will like to either direct the conversation back
to, or portions of the conversation that are links between chat themes. The moderator is not limited to save a chat submission as simply one of predefined set; he has the option of saving the submission as a new type. The moderator has the option of saving the concepts from the chat discussion to either a previously existing concept map or a new concept map. The tools to perform these actions are available to the moderator at his menu bar.

4.4 Database

Each selected text that is highlighted by the moderator and its associated data is stored in a mySQL database. Associated data refers to the person that submitted that text (submitter), the time since the chat discussion was started that the text was submitted (time elapsed), the context of the text (i.e. whether the moderator believes the text is of type brainstorming or a tag etc.).

The database contains 7 tables:

1. Chat Main: contains the title of the concept map and the date each of the concept maps was created.

2. Chat Log: a log of all the conversation types that the moderator has selected.

3. Discussion: This contains all the text that the moderator has selected, the time the text was submitted, the type of the text and the person who submitted the text.
4. Discussion Temporary: This contains similar information as (3), but for a specific chat
discussion. It is the data from this table that is used to make the concept maps. After the
concept maps are created, this table is cleared.

5. Insert Chat: Used to enter labels from the concept map to the chat discussion and to
link concept maps.

6. Other type: used for keeping track of new types that the moderator chooses to save a
conversation.

7. Server Port: contains the port that each server connects to.

The criteria for selecting the database were as follows: the database needed to interact
easily with Java code, it needed to be free, to be able to store a relatively large amount of
data, to have an existing online infrastructure of support that was free and easily
accessible, to have been tested and proven to be robust an secure by others and relatively
easy to install on both a Windows and Unix platform.

For these reasons mySQL was selected as the database. In addition to being free, easy to
install and having an active support community, mySql can be used on multiple CPU’s,
and can work on many different platforms including C, C++, Java, Perl, PHP, Phyton.
The password system is flexible and secure. It allows host-based verification, and
passwords are encrypted one connects to a server. It has an ODBC (Open Database
Connectivity) and has been proven to handle large databases. There have been
documented cases of mySQL databases with 60 000 tables and 5 000 million rows.
MySQL was tested by a commercial memory leakage detector and revealed no memory
leaks. Finally, Clients may connect to mySQL servers using TCP/IP sockets, Unix sockets and Named Pipes.

Using a mySQL database offers the CDA system greater flexibility in terms of scaling.

4.5 How CDA works

CDA uses TCP/IP sockets to communicate. The main chat system consists of three classes: ChatServer, ChatHandler and ChatClient. Instances of ChatServer and ChatRun (this calls ChatClient) are created from command line executions. The ChatServer class is responsible for accepting connections from clients and providing each client with a dedicated thread for messaging. The ChatHandler is an extension of the Java Class Thread. This class is responsible for receiving client messages and broadcasting those messages to other clients.

Running the Chat Client takes either a single command-line parameter -- the server name, or a two command line parameter – the server name and the word 'moderator'. From this, the system looks up the port number that is associated with the server for listening. It makes a socket connection and then opens the frame with the four components. After the user types text into the input region and hits Return, the text is transmitted to the server. The server echoes back to every client that is listening on the server port everything that
was sent by the client. The client displays everything received from the server in the output region.

4.6 Moderator’s viewpoint

The moderator is presented with more menu options than a regular user. It is only the moderator who has the ability to create the concept maps. Having each user create a concept map while simultaneously trying to chat would reduce the effectiveness of a free flowing chat conversation. Additionally, Coterie (Chapter 3, pp ..) demonstrated that when given the same chat logs, humans have differing views on what idea or conversation theme a given contribution belongs to. Thus, if each user was allowed the opportunity to create his own concept map, then for someone who was not present at the conversation and wanted to know what ideas were generated, that person would have to look through all the concepts maps and form his own opinion. This is no better than having the user go through the actual chat logs. While at the end of a chat session, only the moderator will have created the concept map(s), each user after the end of the session can use the moderator’s map to create his own. The difference being that only the link (title, date created) to the moderator’s one is kept on the mySQL database. The actual concept maps (those created by the moderator and those created by other chat participants) are also kept on the server. This adds to the scalability of the CDA system by allowing for the option of having multiple concept maps accessible to everyone if the ‘moderator solely generated option’ is not a viable characteristic.
While the moderator is given the responsibility of decided how the chat contributions are connected to form concepts, he is also disadvantaged as his additional responsibilities reduce the time he has for chatting. So the choice of who is the moderator for a given chat session needs to take into account which of the participants is most familiar with the issues being discussed, who can process information quickly and who can type quickly.

As chatting progresses, it is the moderator who selects portions of text from the chat that he views as relevant. Selecting entails highlighting the relevant section and then choosing from the Save Text As ... menu option the appropriate type to save the text as. The moderator has a list he can save the text as or he can create a new type and color for the text.

After the selection occurs, the selected text changes color to signify that it was selected. The color it becomes depends on the type the text was saved as. The color encoding provides an effective means of informing the moderator what has been archived without having him to go through a lot of operations and presenting the information in an easily understandable format.

Saving a text as a particular type results in that text and its associated information being stored in the database. This is the only additional task that the moderator has to perform while the chat discussion is in progress. At the end of the discussion, the moderator has the option of saving the selected text as an existing concept map or a new one.
ILLIC ts(137) :- How many cats does it take to make a fur coat?

ILLIC ts(156) :- type of coat

TORTUE ts(181) :- age

ILLIC ts(182) :- size

BUM ts(183) :- type of cat

ILLIC ts(237) :- utility of cats used

Fierre ts(238) :- whether only one type cat is used or a variety.
4.7 Regular user’s viewpoint

The following features are accessible to everyone who is currently engaged in a chat conversation. Any user can select to look at a saved concept map. Within that concept map, a user can select a label for a node or a tag and insert that label into the chat discussion.

Another user can select the inserted text to view the concept map. Each user has to ability to correct his spelling mistakes. The user highlights the text he wants to replace and chooses the replace text option. The system checks the highlighted text and the participant that highlighted the text against the list of text and the contributing participants in the database. If there is a joint match the text is replaced.

Each participant can also check to see if a simultaneous chat discussion is also occurring. This is done by creating a ChatClient instance that attempts to connect to the other server/port connections stored in the database. Because of the TCP/IP protocol for the CMA, the system cannot support having to continually listen on other ports for messages.

![Chat Interface](image)

**Figure 3:** Checking to see if another chat session is in progress.
4.8 Concept Maps

The underlying program, used in the creation and display of the concept maps is a jKSImapper. jKSImapper is a client/server graphical implemented using the Java Programming Language. The application was developed by Roberto A. Flores-Méndez as part of his Masters of Science June 1997 Thesis within the Department of Computer Science, University of Calgary.

jKSImapper is modeled after KSIMapper, a C++ based program that allowed a single user to create and manipulate concept maps. The features of jKSImapper that make it a suitable add on to the Chat Decision Assistant includes: it is a Java based program, so can be used on all platforms where a Java interpreter is implemented, a multiple user interface can be added to the system, the program can be imported into a browser using an applet, jKSImapplet and it is open source.
4.9 Windows Manager/Menu Bar Options

The Chat GUI has four (4) pull down menu options:

File, Save Text As ..., Insert Text, and View.

File

Figure 5: File Options

Allows for the opening of previous stored concept maps. Only the moderator has the sub-menu option of saving the chat discussion as a concept map.

Figure 6: List of Chat History
Save Text As ....

**Figure 7: Options to save Text As.**

Only the Moderator’s chat GUI has this option. It is from this pull down menu that the moderator is able to highlight the chat submissions.

**Insert Text**

Allows for the insertion of concept map labels to be inserted as text in the chat GUI text display window. The Replace option under this menu allows for words to be replaced.
View

Shows the concept map that has been linked in the text display region of the chat GUI.

**Figure 8: View Options**

![View Options](image)

**Figure 9: Chat Transcript**

![Chat Transcript](image)
Figure 10: Checking for other Chat Sessions

Welcome to the Chat
ILLIC :- Who wants to repor

Please wait while system checks if any other chat sessions are open.
4.10 Concept Map Menu

Link

Allows for the insertion of concept map labels to be inserted as text in the chat GUI text display window. The Replace option under this menu allows for words to be replaced.

4.11 Implementation Details

The code for the Chat Decision Assistant was written in Java for similar reasons as why the mySQL database was selected. The implementation was divided into three parts:

1. The Front End GUI
2. The Back-end data processing
3. The Concept Map interaction
4.11.1 Front End GUI

The GUI represents a typical interface a text based chat user might encounter. The front end section can be viewed as an actual text based chat system. It is composed of two main subsections:

1. The Client Class

In terms of regular chatting, it also creates instances that act upon echoes from the server. The Client Class consists of ChatRun, ChatClient and ChatHandler.

2. The Server Class

Handles the requests of all the clients that connect to it. It receives broadcasts from each client and echoes the broadcast to a given port on all the client machine.

4.11.2 Back End Data Processing

There is overlapping between this section and the Front End GUI in that action events that occur in the GUI pass control onto the back end data processing classes. The ChatClient class uses a menu-event handler to determine which class control is passed onto. This class (ChatClient) calls Java instances based upon the action events of the menu option selected by the moderator in the Chat Window Manager.

The data processing includes the creation of an analysis window, the storing of selected text, the editing of chat contributions and the checking for other concurrent chat sessions.
2. Analysis Windows

There are two methods used to analyze past chat discussions. One looks at the chat log as a function of time and user contribution, the other as a function of the type of the contribution and the time the contribution was made.

![Chat Concept Map Window](cats2.jcm)

**Figure 11: Analysis Menu**
2.1 TimeLine Window

Displays a timeline graph that shows the type of contribution that each participant made to the discussion. It uses to color coded format to display the information in an easily understandable format. The color format is the same as the one used in the concept windows and in the highlighting of the chat text. Each participant that made a contribution has a timeline in the window.

2.2 Type Window

For each type of conversation it shows when such a type appeared in the conversation and the participant who made that contribution. The color system here is more of an offsetting format where it is used to separate one entry from the other. No new information is added through the use of color here.

These windows allow chat participants and users who were not present during the conversation to determine who spoke the most and who contributed the most ideas of a particular type. This provides two ways of determining who owns the conversation or a particular section of the conversation type so they can know whom to best contact for more information.
4.11.3 Concept Map

Concept Map Window

This controls the appearance of the concept windows, the opening of new concept maps, the linking of concept maps and the saving of altered concept maps.

1. Label Edit Dialog

Manipulates the labels on the concept maps with the text that appears in the chat window. Places the highlighted chat segments selected by the moderator as labels in the concept maps and imports labels from the concept maps into the chat conversations.
Chapter 5

Evaluation of Implementation and Effectiveness

This chapter describes the evaluation of the actual Chat Decision Assistant program that was developed for making decisions based upon chat discussions. The problems that were discussed in Chapter 2 are used as a benchmark to determine the effectiveness of the Chat Decision Assistant.

The key goals of the system were to:

1. create a method of archiving the chat discussions that will encourage members of the community to refer to the archives,
2. the archiving must not be static, in that users can use the archives in communication,
3. the system must aid in decision making with out directing the discussions
4. the system must not hamper the free flowing conversation normally associated with chat programs.

5.1 The Front End System

Except for the moderator’s viewpoint, for the rest of the users the front-end system is mostly similar to text based chatting. The constant needing to have to click to highlight
the text that the user wishes to perform some action upon does reduce the time that users have available for chatting. This makes the option to correct spelling less appealing than simply retyping the word if the user is a fast typist or if the chat channel is receiving a lot of messages very quickly. The ability to correct spelling will be useful to the moderator when he imports the selected chat contributions to form concept maps.

5.2 The Data Processing

The computerized system performs the tasks of taking the highlighted text from the concluded chat session and converting it into a basic concept map where the vertical axis is time dependent and the horizontal axis depends on the type of the contribution. Most of the analysis work is not computerized as it is rather performed by the moderator (a human). It is the human who decides what contributions are worthy concepts and who determines the category that the contribution falls under. It is also the moderator who modifies the computer generated concept map to form the interlinking version that the rest of the community views.

As the data processing (highlighting text, modifying contributions, selecting types and tags) is not visible to most users (except for the moderator), the data processing in the Chat Decision Assistant (CDA) does not affect the chat conversations. Most of the data processing: joint searches on mySQL databases, interfacing with the jkSIMapper is done
after the chat session has been terminated so the computer resources that the processing takes will not affect the chat discussions.

5.3 The Concept Maps

The system is designed to have the concept maps intertwined with future discussions. When a user links a concept map to a discussion, it forces the rest of users to look at that map to understand what was said. In one way this does direct the flow of the conversation in that users take time to look at the map and understand what ideas the map portrays. The users are then more likely to use other ideas from the map in future chat conversations. From the other side of the coin, having all the users look at the concept map jolts their memory of past ideas and allows them to arrive at better decisions. Looking at the issues generated from past discussions also keeps the decision making process on track. Users remember what was said in the past, what are new decisions that need to be made and who was assigned what tasks.

The structure of the concept maps is geared towards assisting the decision making process. The types (nodes and links) of the concept maps are the general categories that decision making falls under.
5.4 The Analysis Windows

There are two analysis windows: the TimeLine window and the Type window. The first shows the type of contribution each user made to a particular chat session; the latter shows whether the chat conversation was more about idea generation or analysis of past projects. The first window is useful for determining who has ownership of chat conversation. One disadvantage of the current system design is that to access the information the user must be logged in to a chat conversation.

This limits the time one can analyze the past discussions. It will be more efficient to have the archive concept maps stored on the web on a common site that is easily accessible to everyone.

5.5 Overall Analysis

The main difference between this chat program and other programs is the emphasis of this program on decision-making. Other differences include, the program is designed for a more restrictive set of users, those who have a common goal and want to use the chat program to work towards achieving that goal. The objective of not impeding the free flowing conversations is hindered by the need to provide a system geared towards decision making. The reasoning behind not impeding the conversation flow is to ensure that the system does not discourage chatters from using it. Given the users whom the
system is designed for place a higher priority on arriving at more effective decisions, the trade-offs in the system design are reasonable.

The moderator is the most affected user of the system. He is less likely than other users from being able to contribute to the conversations, as he constantly needs to be aware of every context that a submission can belong to, and he has the task of selecting the text. The system has no safeguard that prevents the moderator from inserting ideas that were not mentioned in the conversation. However, nodes or links that the moderator creates are different from the ones that originate from the chat discussions in that it does not have the accompanying time stamp. Also, entries that the moderator creates are not linked to the chat database tables, instead they are only stored on the tables associated with the concept map. These entries will not appear in the analysis windows (TimeLine or type window).

Overall, the system achieves at some level the goals set out. Some of the goals, free flowing conversation and conversation geared towards decision-making were implemented in a manner that caused the goals to be conflicting. The next chapter, Chapter 6: Future Work, discusses improvements to the system that will result in a better evaluated program.
Chapter 6

Future Directions of Research

With a global village that is increasingly becoming smaller, there are several ways that the work of this thesis can be expanded. In terms of short-term goals, instead of having the system be run as an application, it could be embedded in a Web page for future development. This could allow the system to be combined with other web based collaborative tools. For example, the system could be combined with existing collaboration based web sites such as the Media Lab Ireland video and messaging board collaboration site designed by Deirdre Butler and Jamie Rasmussen. This will also assist in the addition of asynchronous elements to the synchronous chat. As mentioned in Chapter 5, instead of the current situation where it is only during an existing chat that someone has access to the concept maps of previous chats, it will be more effective if users could log on to a web page to view previous concept maps.

In order to be a true collaborative tool, CDA will need to allow users to interact with other internet based collaborative tools such video conferencing and electronic whiteboards. CDA could have a pop up frame that allows for the attachment of images/drawings. This new window will have a MouseListener class associated with it to incorporate inputs from the mouse for image drawings. This could also be expanded to changing the protocol of the frame to move from a text window to one that allows for URL links to be displayed. The add-on tools could also include an offline ‘post-it’ option.
Before the start of a chat session, each user can compile a list of topics he will like discussed.

Currently, every new option from the Menu bar results in the CDA popping up a new window. It will be more user efficient if instead of having multiple screens, to have one frame with multiple windows. Also the viewing of sub levels in concept maps can be altered so instead of having a new concept map window appear when a linked node is clicked, the current concept map window should expand to show the sub-details. Clicking on the linked node will result in the sub-details disappearing.

Long term goals include making the CDA system compatible with commercial collaborative tools such as NetMeeting. The system should include forums, mailing list archives, a knowledge repository of works in progress, group meetings, records of previous documents, audio and visual media that supports live video conferencing web channel feeds and being able to archive such discussions.
References


Aria

Gesture and Narrative Group –MIT Media Lab
http://avatars.ww.media.mit.edu/avatars

IBM Research – Social Computing Group
Babble

The Internet for Beginners
http://netforbeginners.about.com/library/glossary/bldef-chatroom.htm

Internet Relay Chat (IRC) Help Archive
Frequently Asked Questions
http://www.irchelp.org/irchelp/altircfaq.html

Lucid Chat – The Future of Web Based Chat
http://www.lucidchat.com
The MIT Media Laboratory, Future of Learning  
http://learning.media.mit.edu

The MUD Connector’s FAQ  
http://www.mudconnect.com/mudfaq

TechSoup – The Technology Place for Non-Profits.  
http://www.techsoup.org