Conversation Trees and Threaded Chats

Marc Smith & JJ Cadiz
Collaboration & Multimedia Group
Microsoft Research
One Microsoft Way
Redmond, WA 98052 USA
+1 425 936 6896
{masmith, jjcadiz}@microsoft.com

Byron Burkhalter
University of California, Los Angeles
2201 Hershey Hall
610 Charles E. Young Drive
Los Angeles, CA 90095-1551
+1 408 445 9779
burkhalt@ucla.edu

ABSTRACT
Chat programs and instant messaging services are increasingly popular among Internet users. However, basic issues with the interfaces and data structures of most forms of chat limit their utility for use in formal interactions (like group meetings) and decision-making tasks. In this paper, we discuss Threaded Text Chat, a program designed to address some of the deficiencies of current chat programs. Standard forms of chat introduce ambiguity into interaction in a number of ways, most profoundly by rupturing connections between turns and replies. Threaded Chat presents a solution to this problem by supporting the basic turn-taking structure of human conversation. While the solution introduces interface design challenges of its own, usability studies show that users' patterns of interaction in Threaded Chat are equally effective, but different (and possibly more efficient) than standard chat programs.

Keywords
Chat programs, turn-taking, conversation, computer mediated communication, synchronous communication, persistent conversation, human computer human interaction

INTRODUCTION
Chat is an old and increasingly popular form of computer-mediated communication. Commercial on-line service providers like America Online and non-commercial networks like Internet Relay Chat provide a myriad of chat rooms filled by millions of people daily. Instant messaging programs from AOL, ICQ, Yahoo, and MSN are becoming increasingly popular. 430 million instant messages are exchanged each day on the AOL network, and 330 million are exchanged on ICQ [12]. This form of communication is likely to increase as cell phones and wireless handheld computers make mobile messaging even more prevalent: wherever cell phone short message system (SMS) service is available, its use is rising dramatically. Chat is here to stay.

Although these chat programs are popular for informal interaction, several companies are now bringing chat to the business world [2]. However, chat has not evolved much in the past twenty years and remains poorly suited for holding complex discussions. Innovations in chat have mostly ignored this problem. There have been a number of chat systems released by commercial Internet software companies that have integrated a variety of 2D and 3D graphical representations with standard chat [1, 4, 5, 14, 23]. However, few have altered the way chat organizes people's exchanges of messages in a positive way, making chat even less easy to comprehend. The recent explosion of "Instant Messages" and "Buddy Lists" has not changed the underlying structure of chat either.

In this paper, we will discuss the core problems we see in chat and describe the ways this guided our design of Threaded Chat. In addition we report the results from a lab study that tested the usability of Threaded Chat in contrast to standard forms of chat with eighteen small groups engaged in a decision-making task. We discuss the challenges raised by the design of Threaded Chat and suggest future directions for improvement of systems to support persistent computer-mediated interaction.

COMPARING CHAT AND SPOKEN CONVERSATION
Chat may be a form of computer-mediated communication that closely resembles spoken interaction, but in contrast to spoken interaction, chat is poor at managing interruptions, organizing turn-taking, conveying comprehension, and resolving floor control conflicts. Studies of chat from a variety of fields (including sociology, communication, CSCW, HCI, and linguistics) share a focus on the challenges and ambiguities chat introduces into the normal mechanisms of social interaction.

Conversation Analysis (CA)—the sociological study of the structures of ordinary face-to-face and spoken interaction—is of particular value when seeking ways to improve chat. CA's study of naturally occurring conversation reveals that people use a suite of fine tuned, ordinary techniques for maintaining coherent and understandable spoken conversations. Spoken conversations have turn and response structures governed by a set of simple rules that organize how groups of people exchange turns of talk.

Sacks et. al. [18] argue that turns are valuable commodities that require an orderly allocation system:

> For socially organized activities, the presence of 'turns' suggest an economy, with turns for something being valued—and with means for allocating them, which affect their relative distribution, as in economies.

Using simple turn-taking rules, people are able to sustain spoken conversations across a wide variety of topics where there is almost always one party talking at a time. Interruptions and overlaps do occur but are brief, and transitions between speakers commonly occur without gap or overlap [18]. In contrast, in its most common form, chat organizes turns in order of their arrival at a central server, not in the order of turn and response in which they were constructed. This undermines the techniques people use for organizing coherent conversations [9]. The result is an inclination for confusing exchanges of short messages in ambiguous order. This makes chat a poor decision-making tool and knowledge store and reduces its value for meetings and presentations of detailed ideas.

Computer-mediated conversation has the potential to transform the constraints of the economy of spoken interaction in more positive ways. Our inability to listen to two or more people speaking at the same time for very long limits the number of possible turns available in any spoken conversation. In contrast, chat may be less restricted than spoken communication since more than one person may construct a message at the same time, and reading can be quicker than listening. Nonetheless, turn-taking systems for spoken discussions allow for more coherent and productive conversations than standard chat programs. Thus, the properties of spoken conversation systems offer guidance for the design of text chat.

CA directs attention towards improving the turn-taking system used in the exchange of chat messages. Threaded Chat presents a possible solution to this problem by supporting a synchronous form of the turn-taking structure found in asynchronous threaded discussion boards like Usenet. Systems like Usenet and a vast number of discussion boards on web sites allow for the creation of extensive discussion trees composed of message ("post" or "article") turns and responses linked together. These systems have predominantly been used as a form of asynchronous interaction in which delays of hours or days between turns and responses are common. While these systems suffer from problems of their own [13], discussions of complex ideas can be developed over time with responses clearly linked to the messages they are in reply to. In Threaded Chat, we have modified this structure to make it more accommodating to both synchronous and asynchronous use.

FIVE CORE PROBLEMS WITH TEXT CHAT

Research rooted in the sociological study of conversation has identified and addressed some of the major issues with standard chat programs [7, 8, 9, 15, 21]. These findings lead us to identify five main flaws in existing chat systems:

Lack of links between people and what they say

Chat programs present each participant's messages in a way that makes it hard to differentiate speakers. The high turnover of participants in many chat rooms further aggravates this problem. Many systems address this issue in one form or another. Some chat clients provide ways of associating a color or font with particular people. More recently, systems have focused on awareness of presence of people in the room [3, 20], representations of the timing of the conversation [21], and improved visualization of conversations [20].

No visibility of listening-in-progress

In chat, participants do not receive moment-by-moment information about the reaction of those who are listening to them. This means that turns cannot be altered as they unfold, increasing the likelihood that they will be misunderstood or taken in the wrong way. Without indications of listening, chat systems loose a great deal of their sense of social presence.

Some experimental systems have addressed this issue. Erickson et al.'s Babble [4] addressed this problem by presenting a "social proxy", a graphic design that represented the activity of people with the application. This allowed people to have an intuitive sense of who was recently active but lacked the granularity to present reactions to turns-in-progress.

Lack of visibility of turns-in-progress

Chat systems only transmit turns when users press the ENTER key. While some systems do transmit messages keystroke-by-keystroke (i.e. the Unix program "Talk") most do not. As Garcia notes, the result is that the process of message production is separate from message transmission. Chat is not truly synchronous: it has a sporadic rhythm in which fully formed turns pop out in a single moment instead unfolding in real time. Chat lacks the "mutual availability of utterances-in-production" [8].

In contrast, the moment-by-moment surveillance of others in spoken conversations allows people to be highly sensitive to small variations in timing. For example, when declining an invitation or disagreeing with another's assessment, people will often slightly delay the beginning of their turn. The delay projects a dispreferred response [15] (a response the listener would not like), allowing the original assessor to downgrade or alter the assessment in order to maintain agreement. People are able to connect turns so quickly and assess the gaps between them because speakers project where their turns are heading and listeners recognize those projections as the talk unfolds.

Delays in chat resulting from typing difficulty or the other user leaving the room can easily be misinterpreted as a dispreferred response. Furthermore, delays encourage users to type additional turns (which may modify their initial turn or start a new topic of conversation) instead of waiting. Garcia [8] found that timing and sequencing distortions introduced by standard chat systems meant that a significant portion of chat turns were used to clear up confusion caused by prior turns.

Vronay's Flow Chat [21] explicitly presented the stream of time and the resulting interleaving of turns of chat. Flow Chat placed each user's text on a separate vertically stacked parallel track. While text entered by the user was not displayed until the turn was completed a colored band was extruded from the right side of the display to indicate when the user began typing and how long they had been composing the message. Once entered the text was displayed in the color bar, which then continued to slide towards the left of the display on its track. While this clarified the sequential ordering of turns, it did not provide any other way to indicate a link between two turns. In large groups this means that links between turns separated by many tracks were difficult to associate.

Viegas and Donath's Chat Circles [20] approach this from a different direction. Chat Circles presents each user as a colored circle that expands with the amount of text entered by the user. Circles then slowly shrink in size as the text fades. The timing of turns is thus visible and turns-in-progress are presented as expansions in the size of the circle. This view of the conversation lacks a historical component as turns evaporate over time. As a result the application has an alternative historical view, which visualizes the conversation along a vertical time line cross marked with lines indicating the timing and size of each user's turn. This is in many ways an alternate form of Vronay's Flow Chat that shares its limitations.

Microsoft's MSN Messenger is one commercial product that partially addresses the problem of seeing turns-in-progress. When others are typing, "[name] is typing a message" appears at the bottom of the window. Although this alleviates some of the problem by providing a binary indicator of typing, it does not entirely solve the problem because users cannot see exactly what others are typing until the ENTER key is pressed.

Lack of control over turn positioning

Much of the work in conversation coordination relates to shaping a turn's meaning based on its location. However, the techniques used to accomplish this in spoken interaction are undermined in chat conversations [9]. Standard forms of chat position turns based only on the time that the ENTER key is pressed, which often ruptures the links between turns and their replies. "Participants in QS-CMC cannot assume that their attempts to be a 'first poster' will result in the message they are typing being placed adjacent to its intended referent," writes Garcia [8].

In standard forms of chat, ownership of the floor is only known when a turn is completed, at which point a race begins to finish one's own thought, which is newly fitted to the recently emerged turn. This twisted set of conversational rules has two ramifications: first, one can only begin to fit a "next" turn after the last turn has been displayed in its entirety, and second, there is a preference for short turns because one must press the return key in order to secure the floor. Therefore, extended turns, which can allow more complex material to be discussed, are much less frequent.

For example, consider the following chat interaction:

```
1 Larry:  boy do we need to work on our
          interview skills....
2 James:  who's conducting the
          interviews, anyway?
3 Scott:  Yes
4 James:  okay...
5 Larry:  All of us
```

Notice that James and Scott are entering both turns simultaneously. Each turn is fitted to Larry's initial turn. Although Scott's turn "Yes" appears immediately after James' turn "who's conducting the interviews, anyway?" it obviously does not fit as the next turn. Similarly, Larry's turn "all of us" follows but does not fit the prior turn of "okay…". The only way users can make sense of the turn is to scroll up and find a candidate "prior turn." That people can do this is itself interesting, but the procedure is time consuming (and the conversation continues while this is done). The result is that transcripts of chat conversations are often confusing and demand significant effort to read.

Babble [4] addresses this by designing for an expectation of slower interaction rates than typically found in chat. The slower rate allows users to have greater certainty that their turn will occupy the position it was crafted for. As a result, short expressions of concurrence (ex. "I agree", and "yes") are possible and meaningful. Sequencing problems do sometime occur, however, and are likely to increase if Babble is used more synchronously.

Lack of useful recordings and social context

Chat rooms are social spaces that never develop a social history [3]. In practice, most chat rooms are not publicly persistent: their content evaporates as soon as it scrolls out of each user's history buffer. This lack of persistence means that most chat spaces do not accrete a social history. Groups do use other media (for example, web pages) to create durable artifacts of their interaction, but the chat room itself does not change as a result of the activity within it. Even if logs are maintained, as noted above, the resulting transcript is often nearly unintelligible.

This usually is less of a problem during the conversation than several days or months later when one tries to review chat logs. For instance, when a chat conversation occurs, if two turns appear within a tenth of a second of each other, it is probably clear to an attentive participant that the second turn was not intended to be a reply to the first. However, timing cues are missing from most history logs. Thus, ruptured and jumbled turn sequences make the conversation log ambiguous and unreliable as records. (This problem can be addressed by including timestamps with chat logs, but reconstructing the events of a chat room using timestamps is tedious.)

This has two implications. Having no useful recordings of chat conversations is a significant obstacle in workgroups and business environments, particularly when used in decision-making processes. It also means that chat programs demand full immersion to remain comprehensible to their users. When users look away or try to maintain peripheral awareness many find it difficult to catch up with conversations.

THREADED CHAT

Threaded Chat addresses the problems of confusing history logs, lack of social history, and the rupture of turn sequences in standard chat rooms. Threaded Chat departs from traditional chat in a number of ways by bridging the gap between threaded asynchronous discussions and synchronous chats. The Threaded Chat user interface is displayed in Figure 1. All chat turns are structured as a tree, similar to the Microsoft Windows Explorer interface to the file system on a computer's hard disk, and similar to an idea proposed by Herring as a way to make chat conversations more coherent [9]. The key element of this structure is that turns are organized into turn and response structures called *threads* that can grow to any size. Thus, proper use of Threaded Chat eliminates the possibility of ruptured sequences of turns: turns are linked directly to the turn they are intended to respond to. Even if a turn is

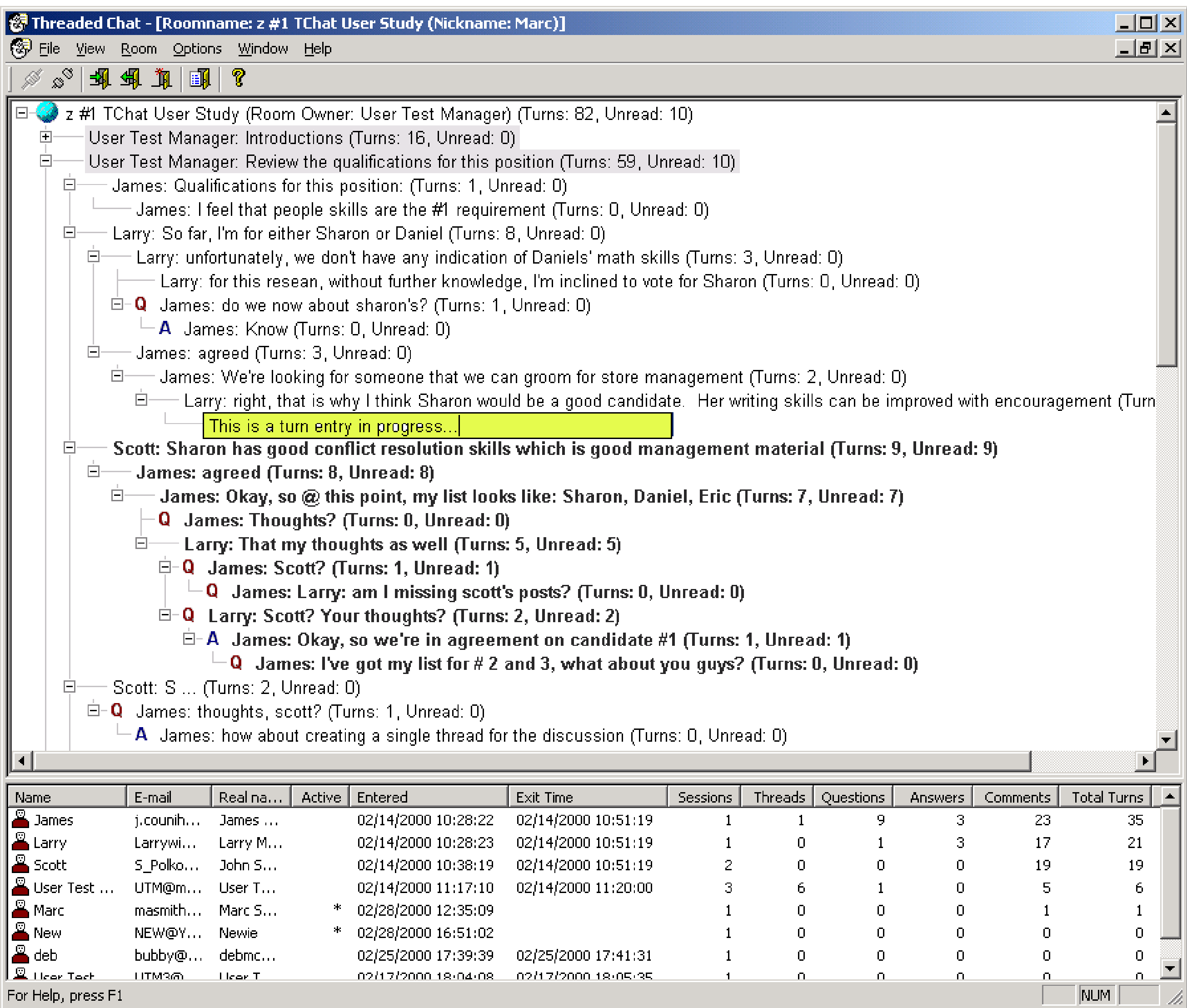

Figure 1: The Threaded Chat user interface. Users chat in the top portion of the window while participation information is displayed at the bottom.

misplaced, it can be dragged and dropped to the correct location. Turns can also be edited or deleted.

To chat, users click on the turn they want to respond to and being typing. Pressing return completes the turn. When a user begins to enter text, their name and a placeholder message ("Entering Text") appears to all other users. When the return key is pressed, the entire message becomes visible to everyone.

As turns are entered, they are displayed to other users in a bold font. Over time, the font fades to gray so that most recently added turns stand out clearly. This feature is especially important since Threaded Chat does not structure turns in order of arrival (a point we return to below). Turns are unbolded and marked as read when clicked on, replied to, or cursored over with the arrow keys. As a turn is replied to, the count of the number of replies and unread turns beneath it are displayed.

Selecting the room node at the top of the chat room and entering text creates a new top-level thread, which is highlighted with a colored background. Top-level turns are typically the major topics of conversations, thus they are distinguished from other turns.

The tree structure of Threaded Chat provides users with the ability to collapse any branch of the conversation if they no longer wish to pay attention to it. For example, users may collapse discussion branches that no longer concern them or that have come to a conclusion and are no longer pertinent. If additional turns are added to a collapsed thread the count of unread child turns is incremented.

The bottom of the Threaded Chat window contains information about the conversation participants. Information includes time of entry, number of entries (labeled "sessions"), and time of exit. Basic statistics about the number and types of turns are also displayed. These statistics persist from session to session, and users remain in the list even when they are not present (although they are marked as not currently active). This information is useful for providing a sense of history and context for the room.

Threaded Chat automatically labels turns that are likely to be questions or answers. If a question mark is found in the text of a turn, the turn is tagged with a "Q". All replies to questions are tagged with an "A". Numbers of question and answer turns are tracked in the social accounting pane.

Turns can be edited, deleted, or dragged and dropped to different places in the tree. Although this can be a helpful feature, it also raises the possibility of abuse. Thus, each Threaded Chat turn has permission properties based on an extension of the Unix user/group/world model. These permissions are accessed by right clicking on a turn and allow a turn's author to determine who can see the turn, reply to it, delete it, and extend the turn's permissions. Only authors of a turn can edit the turn's text.

Turn authors are also the only people who can modify a turn's permissions, although owners of turns higher up in the tree may override the rights. For example, if person B replies to person A and specifies a set of rights on the reply, person A could override person B's permissions by specifying rights on the original turn. Users retain the power to override permissions of the turns that are replies to their turns, including the power to delete or move the entire thread to another location. The first person to start a big conversational branch wields significant power over it. Using Threaded Chat's permissions, it is possible to have a private chat in the middle of a public room, or to have a public discussion with a select group without possibility of interruption from others. It also means that users can enter a turn and determine who may see and reply to the text.

TESTING THREADED CHAT

Given that Threaded Chat is designed to address some of the key problems with standard chat, we conducted a user test to see if the design was successful.

Specifically, because proper use of Threaded Chat guarantees that turns will always be placed in their intended context, we expect that:

- Threaded Chat will support better topical coherence than standard chat. Users should be able to maintain coherent sequences of conversational turns more easily.
- Transcripts of a Threaded Chat discussion will be easier to comprehend than transcripts from a standard chat room.
- Turns will be longer than those found in standard chat rooms. People will know that they do not have to rush to get their turn in before someone else types a message, thus Threaded Chat turns should use more words and/or characters.
- Fewer repair statements will be made. As a result, we expect participants will produce fewer turns using Threaded Chat when compared to plain chat.
- There should be a more balanced level of participation among people using Threaded Chat. With standard chat programs, people frequently abandon their turns when they are not able to finish before others enter turns that change the conversation context. Threaded Chat guarantees context for a turn, thus there's no reason to abandon turns. We believe this effect will be especially true for slower typists.

Experimental Methodology

70 participants were recruited for a lab study to test Threaded Chat. Participants were grouped into eighteen teams of four; however, due to no-shows, eleven groups had only three people. Participants received a free Microsoft software product for their time.

All participants had used a chat program at least once in the past year, were comfortable with typing, had graduated from high school, and were at least intermediate computer users. Participants were recruited such that the pool was diverse in terms of age, gender, and occupation. The pool had 38 men and 32 women. The average age was 39 with a standard deviation of 10.

Participants used three different chat programs for this study. Threaded Chat and a "standard" chat program were used. In addition, this study was combined with a study of another experimental chat program, LeadLine. (A report of the LeadLine study is available in these proceedings [6]). The order in which the three chat programs were used was counterbalanced to minimize order effects.

Participants were told they were employees for the same company and had recently interviewed three candidates for one job opening. Their task was to chat with each other for 20 minutes and then, as a group, rank the candidates in order of hiring preference. This task was repeated three times, each time using a different chat program, a different set of candidates, and a different job position. In each case, participants were given unique information about the candidates so no single participant could correctly rank the candidates without chatting with other group members.

When using the Threaded Chat program each group started with a room populated with six initial threads:

```
Introductions
Review the qualifications for this position
Discuss candidate #1
Discuss candidate #2
Discuss candidate #3
Final decision: Who should we hire?
```

Although these threads were made available as guides for the discussion (similar to a agenda for a business meeting), users could (and did) ignore them if they wished.

USER STUDY RESULTS

After each session, participants answered a variety of questions about their reactions to the chat program they used. On all the measures except one, Threaded Chat was rated significantly worse than the regular chat program (Table 1). To a certain extent, this was not surprising given the early stage of the prototype: some basic user interface issues had not yet been resolved (for example, lines that were longer than the screen width did not automatically wrap to the next line). However, the core concept of chatting with threads was functional, thus these data indicate that participants did not think highly of Threaded Chat. Reasons for this reaction are explored further in the participant comments section.

Task Performance

Despite the lower subjective ratings Threaded Chat received in contrast to standard chat, the study showed that users quickly adapted to the new interface. Performance on the hiring task did not differ significantly between programs. Each hiring task was designed such that there was a correct solution, and each set of candidate rankings was assigned a score relative to its distance from the correct solution. The highest possible score for each task was 5 points. Threaded Chat groups had an average score of 3.7 while plain chat groups had an average score of 3.9. This difference was not found to be significant, even when taking into account various demographic variables such as typing speed, level of education, and experience with chat programs.

Levels of Participation

Even though scores on the task were equivalent for each chat program used, Threaded Chat did affect the processes used by groups to reach their decisions.

Groups that used Threaded Chat took fewer turns than in the regular chat program. Threaded Chat rooms had an average of 21.7 turns, while the regular chat rooms had an average of 34.7 turns, which is a significant difference ($t(25.5) = 3.7$; $p = 0.001$). In a regression equation controlling for various demographic variables, the use of the Threaded Chat program was the strongest predictor of the number of turns taken, accounting for 28% of the variance ($t = -5.2$, $p < .001$).

Of course, it could be hypothesized that fewer turns were taken in Threaded Chat because people took longer turns. However, this was not the case. The average standard chat turn was 7.3 words long while the average Threaded Chat turn was 7.6 words long, which is not a significant difference ($t(3020) = -1.3$; $p = 0.205$).

	First sessions only				All sessions	
	Threaded Chat (n=6)	Standard Chat (n=6)	p	t / df	Threaded Chat (n=19)	Standard Chat (n=19)
Level of satisfaction (2 questions)	4.0	5.3	0.001	4.5 / 9.7	3.9	5.7
Perceived quality of discussion (7 questions)	4.5	5.2	0.010	3.4 / 7.8	3.9	5.7
Perception of usability of program (3 questions)	4.3	5.9	0.001	6.2 / 9.9	3.6	6.3
Perceived quality of decision process (5 questions)	5.5	5.0	0.345	1.0 / 9.4	4.8	5.8

Table 1: Results from post-session surveys. Possible scores ranged from 1 to 7 where higher scores are better. On the left are data examining only the first sessions (where no order effect is possible). On the right are data from all sessions. Significance statistics are not reported for the data from all sessions due to the possibility of order effects.

	Chat program	Mean	Std Dev	Median
Slow typists	Standard	31.2	13.9	28
	Threaded	20.5	7.2	19.5
	Total	**26.1**	**12.3**	**24**
Fast typists	Standard	36.7	16.2	32
	Threaded	21.8	8.4	20
	Total	**29.3**	**14.9**	**25**
All Typists	Standard	34.1	15.1	30
	Threaded	21.1	7.8	20
	Total	**27.6**	**13.6**	**24**

Table 2: The number of turns taken by slow and fast typists in the different chat rooms.

It is possible that Threaded Chat reduced the ambiguity introduced by standard chat, thus allowing people to enter fewer, more coherent turns whose meaning was partially derived from their parent turn. While the frictions imposed by the Threaded Chat interface may have simply been a drag on the speed of participation, the equivalent scores on the task show that Threaded Chat users were equally able to complete their task using fewer turns.

Balance of Participation

We also examined the question of whether there was a more equal level of participation among group members in the different types of rooms. We used the standard deviation of number of turns taken by the people in each room as a measure of equal participation. If everyone in a chat room took the same number of turns, then a group would have a standard deviation of zero.

Threaded Chat rooms had a standard deviation of 2.9 while standard chat room had a standard deviation of 3.9, which is a significant difference ($t(33.7) = 2.7$; $p = 0.01$). Thus, there was a more balanced level of participation in Threaded Chat rooms.

The next logical question is whether the more balanced level of participation in Threaded Chat was due to slower typists generating more turns, faster typists generating fewer turns, or a combination of both. Based on typing speed, we split all the participants at the median into two groups. Table 2 displays the number of turns that were made by fast and slow typists in each type of chat room. These data show that all typists made fewer turns, but the effect was greater for faster typists.

There are two possible explanations for this phenomenon. First, the method of interacting with Threaded Chat may significantly diminish the advantage that fast typists normally have in standard chat rooms. In a standard chat room, the interaction loop is: read new text, type, press return, read new next, type, press return, and so on. With Threaded Chat, reading new text takes longer because it appears in various locations. In addition, the interaction loop is: read new text, take hand off keyboard, move pointer to turn to reply to, click, type, press return. Thus, typing speed may no longer be the most important variable for rapidly generating turns in Threaded Chat room (which could be verified with a GOMS model [11]).

Second, fast typists may not need to generate more turns. If Threaded Chat significantly reduces confusion resulting from ruptured sequences of turns, then the large number of repair turns documented by Garcia [8] are no longer necessary. However, when we attempted to code turns as being repairs or not, we found very few repair turns in both standard chat and Threaded Chat rooms (only one or two repairs were found in the initial eight rooms examined). When reading the transcripts, it occurred to us that the low number of repairs may have resulted from the exclusive focus on the single task: The conversation was typically composed of only one conceptual thread whereas normal chat rooms are substantially multi-threaded. Thus, we may not have designed a task in which participants could benefit considerably from Threaded Chat.

Use of Thread Structures

Although groups focused on only one conceptual task, they still made extensive use of the thread structures supported by Threaded Chat. The average turn was 3.79 turns deep, the deepest one growing to a depth of 21 turns, with an average depth of 10 turns. Although they are reproduced too small to be read, the overviews of rooms created in three of the trials illustrate the range of variation in the ways threading was used (Figure 2).

Examination of the room logs shows that participants expected themselves and others to properly place turns, something that did not occur at all in standard chat rooms. Thus, a norm for topical coherence was supported:

```
User A: Hi Glen how are you doing
    User A: Hi Glen how do you feel about Joyce
    and her abilities?
    User B: Oops. Should have said that down here.
    Forgot to click on the question.
```

Participants used the thread structure to create extended turns by replying to themselves in a series of turns. This was not the intent of the feature, however it was frequently employed:

```
User C: These are my two choices also and in that
order
    User C: I think we have to eliminate Joyce
    because of her lack or visual graphic arts
    level
        User C: My final is Steve, Linda and then
        Joyce.
```

PARTICIPANT COMMENTS

Although groups using Threaded Chat were able to complete the task as well as groups using standard chat, data from the subjective ratings table indicate that Threaded Chat still has room for improvement. In this section, we examine participant comments to discuss this issue as well as ways in which using Threaded Chat differed from using standard chat.

First, user feedback identified that having no single point of focus was the most significant problem for Threaded Chat.

"…confusing… knowing where to click" (User 1, Threaded Chat trial)

"…unclear where to place comments" (2, TC)

"I was looking all over the screen to see what was being said next" (3, TC)

"Difficult to follow the discussion… Had to scroll too much" (4, TC)

"Difficult to track new messages" (4, TC)

"…cursor seemed to jump around." (29, TC)

"I had to move my eyes all over the screen…" (41, TC)

Standard chat has a single point of focus for new material, which allows users to fix their gaze at the bottom of the screen to guarantee not missing anything new. But in Threaded Chat, new material can blossom anywhere in the conversation tree, requiring frequent scanning to search for new turns.

Users had suggestions for addressing this problem:

"Add some sound so you know when you have new incoming text" (5, TC)

The most frequently requested feature was color-coding for each user to help differentiate speakers.

"Make each person's text a different color" (5, TC)

Despite these problems, users reported that Threaded Chat helped them sustain their conversations more than they could in standard chat, which supported their decision making process.

"For the task… this was the strongest tool." (43, TC)

"I liked being able to follow a discussion thread. It helped keep the idea linear." (4, TC)

"It was easier to answer questions and reread posted information." (9, TC)

"Organizing the information into conversations helped in the decision making process." (21, TC)

"I think it's easier to choose and see what you're responding to with threaded chat." (22, TC)

"…as we used it, we got better, it did allow us to stick to certain subjects." (35, TC)

"…easy to review what has been said about a certain candidate without having to scroll around looking for it." (39, TC)

"I liked being able to manage the screen – close up sections and reopen them. I liked being able to edit messages after they were sent, and move them to other more appropriate sections. It was much different from the noisy conversation style of the plain chat…" (53, TC)

In contrast, when they used the non-threaded versions of chat they reported difficulties managing the conversation:

"…difficult to follow conversation threads when multiples are going." (45, Lead Line trial)

"topics got intermixed…that was confusing." (53, LL)

"if you are not paying attention, you get lost easily" (25, Standard chat trial)

Given the nature of the task, many users felt that access to the history log was essential and that it was poorly organized in standard forms of chat.

"I wanted at times to review what was said earlier in the discussion, but that line had scrolled off the screen and it would have been difficult to keep up with the current discussion and review previous comments." (1, LL)

"…to remember the chat, you have to keep paper notes, the threaded chat was easier in that aspect…" (17, LL)

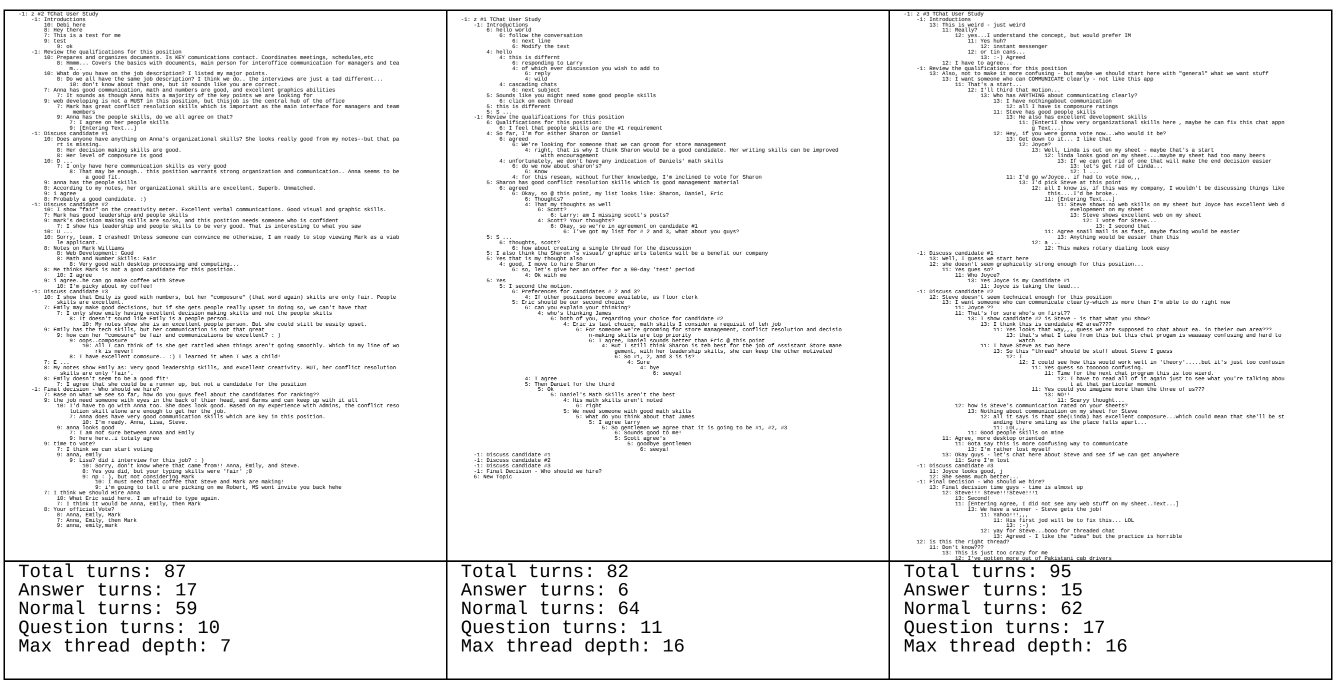

Figure 2. Variations in the thread structures generated in user trials

CONCLUDING REMARKS & FUTURE RESEARCH

Chat has been used for years, but its basic interface remains relatively unchanged. Threaded Chat addresses some of the problems with the standard chat interface, especially as they pertain to business applications.

Our user test of Threaded Chat demonstrated that people could easily adapt to its interface. Although their subjective ratings of the system were low, they were able to complete the task just as well as they could with standard chat. However we found many areas in which Threaded Chat could be improved.

Improving the usability of Threaded Chat requires further research to overcome significant obstacles. How can the interface present new material recently added to distant branches of the tree be presented? How can the interface minimize the jumpy quality of the display as the turns are added at various locations throughout the tree? We are investigating the application of a "thread ruler" that would allow users to reorder threads and provide "limits" that manage the real estate devoted to each threads. In combination with automatic branch control that closed up the oldest turns in each thread branch, the problems associated with attention management may be significantly mitigated. Other directions for development include the creation of alternative views of chat that highlight other dimensions of the conversation, such as its temporal order. In addition, further research questions still exist. The effect of Threaded Chat on the number of repair statements could not be answered by this study, and we also have not tested the readability of the resulting Threaded Chat transcripts.

ACKNOWLEDGMENTS

Special thanks to Jyotindra Tandel for implementing Threaded Chat. Shelly Farnham and Debbie McGhee were very helpful with the planning and execution of this study. We would like to thank Ken Bachman, Megan Hazen, and the rest of the Microsoft Usability Test Coordination team for recruiting and scheduling the participants for this study. We would also like to thank Doug Tutmarc and the Microsoft Usability Setup Operations Support team for their assistance with the use of the Microsoft usability labs.

REFERENCES

1. ActiveWorlds, http://www.activeworlds.com
2. Derfler, F. Instant Messaging: Chat Goes to Work. *PC Magazine.* 19(2), January 18, 2000, 82-101.
3. Donath, J., Karahalios, K., Viegas, F. Visualizing Conversation. *Proceedings of HICSS-32*, Maui, HI, January 5-8, 1999.
4. Erickson, T., Smith, D., Kellogg, W., Laff, M., Richards, J., and Bradner, E. Socially translucent systems: Social proxies, persistent conversation, and the design of babble. *Proceedings of the 1999 ACM Conference on Human Factors in Computing* (CHI 99).
5. Everquest, http://www.everquest.com
6. Farnham, S. D., Chesley, H. McGhee, D., & Kawal, R. Structured On-line Interactions: Improving the Decision-making of Small Discussion Groups. In Proceedings of CSCW 2000, Philadelphia, December 2000.
7. Garcia, A., and Jacobs, J. The eyes of the beholder: Understanding the Turn Taking System in Quasi-Synchronous Computer Mediated Communication. *Research on Language and Social Interaction,* Vol. 32, No. 4, 1999, 337-367.
8. Garcia, A., and Jacobs, J. The Interactional Organization of Computer Mediated Communication in the College Classroom. *Qualitative Sociology*, Vol. 21, No. 3, 1998, 299-317
9. Herring, S. Interactional Coherence in CMC. *Journal of Computer-Mediated Communication*, Thomas Erickson (ed.), Vol. 4, No. 4, June 1999.
10. Hill, W., and Hollan, J. History Enriched Data Objects: Prototypes and Policy Issues, *The Information Society*, Volume 10, pp. 139-145.
11. John, B., and Kieras, D. The GOMS family of analysis techniques: comparison and contrast. *ACM Transactions on Computer-Human Interaction*, 3, 4 (December 1996), pp. 320 – 351.
12. Kobielus, J. Instant Messaging. The Burton Group Network Strategy Report, 1. October 1999.
13. Kollock, P., Smith, M. Managing the Virtual Commons: Cooperation and Conflict in Computer Communities. *Computer-Mediated Communication*, S. Herring (ed.). Amsterdam: John Benjamins, 1996.
14. Microsoft V-Chat, http://www.microsoft.com
15. Pomerantz, A. Agreeing and Disagreeing with assessments: some features of preferred/dispreferred turn shapes. In John Heritage and J. Maxwell Atkinson (eds.), *Structures of Social Action: Studies in Conversation Analysis.* Cambridge. 1984. 57-101.
16. Reid, E. Electronic Chat: Social Issues on Internet Relay Chat. *Media Information Australia*, 67, 1993, 62-79
17. Rintel, S., Pittam, J., Strangers in a Strange Land: Interaction Management on Internet Relay Chat, Human *Communication Research.* Vol. 23, No. 4, 1997, 507-534
18. Sacks, H., Schegloff, E., and Jefferson, G. A simplest systematics for the organization of turn-taking for conversation. *Language*, 50(4). 1974.
19. The Palace, http://www.thepalace.com
20. Viegas, F. and Donath, J. Chat Circles. *Proceedings of the 1999 ACM Conference on Human Factors in Computing* (CHI 99).
21. Vronay, D., Smith, M., and Drucker, S. Alternative Interfaces for Chat. *Proceedings of the 12th Annual ACM Symposium on User Interface Software and Technology* (UIST 99).
22. Wiemann, J., and Knapp, M. Turn-Taking in Conversations, *Journal of Communication*, 25, 1975, 75-92
23. WorldsAway, http://www.worldsaway.com