Understanding Sequence and Reply Relationships within Email Conversations: A Mixed-Model Visualization

Gina Venolia and Carman Neustaedter

September 23, 2002

Technical Report
MSR-TR-2002-102

Microsoft Research
Microsoft Corporation
One Microsoft Way
Redmond, WA 98052

Understanding Sequence and Reply Relationships within Email Conversations: A Mixed-Model Visualization

Gina Danielle Venolia
Microsoft Research
One Microsoft Way
Redmond, WA 98052 USA
+1 425 703 2891
ginav@microsoft.com

Carman Neustaedter
University of Calgary
Department of Computer Science
Calgary, AB, T2N 1N4 Canada
+1 403 210 9404
carman@cpsc.ucalgary.ca

ABSTRACT
It has been proposed that email clients could be improved if they presented messages grouped into conversations. An email conversation is the tree of related messages that arises from the use of the reply operation. We propose two models of conversation. The first model characterizes a conversation as a chronological sequence of messages; the second as a tree based on the reply relationship. We show how existing email clients and prior research projects implicitly support each model to a greater or lesser degree depending on their design, but none fully supports both models simultaneously. We present a *mixed-model* visualization that simultaneously presents sequence and reply relationships among the messages of a conversation, making both visible at a glance. We describe the integration of the visualization into a working prototype email client. A user study indicates that the system meets our usability goals and verifies that the visualization fully conveys both types of relationships within the messages of an email conversation.

Keywords
Computer-mediated communication, asynchronous communication, information visualization, thread, email, electronic mail, Usenet newsgroup, instant messaging, chat, user study

INTRODUCTION
The mismatch between the user interfaces for email clients and user needs for handling email has been documented numerous times [1, 9, 20, 4]. This disparity has stimulated proposals for a plethora of client user interface design changes. One recurrent theme is that messages should be viewed as elements of a *conversation* rather than as independent elements [20, 12, 11]. A conversation, also known as a *thread*, is typically defined as the tree of messages that grows with the *reply* operation.

Several plausible benefits may result from viewing messages as conversations. First, displaying a message along with all the related messages provides *better local context*, which can help one better understand the meaning of the message. Although this context is preserved to a limited extent by current email programs when they automatically include the text of the original message when replying, this method breaks down when a message receives multiple replies, creating a complex, branching reply tree. Subsequent replies provide additional context but these are not captured by quoting.

Second, by making the conversation the main unit of display, more items can be shown at the same time, providing *greater global context*. By collecting messages into conversations, sets of messages that normally would have been displayed on several lines can be displayed on just one line, allowing more conversations to be viewed.

Third, when conversations are presented as units in the user interface, valuable *conversation operations* can be provided. For example in current systems, if five messages that are all part of the same conversation are received, the user has to perform five sets of mouse and keyboard actions to handle the messages (read, file, delete, etc.). However, if the five messages are grouped together as a conversation unit, the user needs to perform only one set of mouse and keyboard actions. Although this may seem like a small benefit, multiplied over a large number of email messages, the benefit may translate to a significant reduction in user effort. In addition, higher-level operations are also possible. For instance, if one starts receiving messages in a conversation that is not of interest, one could opt out of the conversation, deleting all its current *and future* messages.

In this paper, we concentrate on a visualization and user interface that supports the first of these motivations. We describe a design supporting the second, and don't touch on the third. In the next sections, we describe several existing product and research user interfaces that organize email and similar communication modalities into conversations. We then describe two ways of thinking about conversations that are implicit in those designs, and observe an empty niche: fully supporting both models. We then present a *mixed-model* visualization intended to fill that niche, and its surrounding user interface. Finally we describe a user study on the usability of the interface by novices and the degree to which the visualization actually fills that niche. We finish with suggestions for future research in the area of designing user interfaces to support email conversations.

CONVERSATION INTERFACES

There have been many interfaces proposed and deployed for viewing turn-based, tree-structured conversations. Such conversations are a common feature between email and Usenet newsgroup postings, so in this paper we will look at the work in both of these fields. Other research has examined visualizations and user interfaces for turn-based communication where conversations are not a prominent feature [19, 17, 20]. We shall focus on prior work that emphasizes conversations.

Existing Tools

Many widely-deployed tools for dealing with email or Usenet newsgroup postings can display messages as conversations to a greater or lesser extent. Virtually all Usenet newsgroup clients show messages in a tree structure. Microsoft Outlook Express 6 is typical of these interfaces: messages are listed in one pane, and the selected message is shown in another. Root messages (those that are not a reply) appear in the list pane at the top level of indentation; replies appear indented under their parent message, in chronological order. As this layout is recursive, the result is a forest of conversation trees.

Most email clients organize messages chronologically by default without regard to conversation. Some email clients, e.g. Microsoft's Outlook XP and QualComm's Eudora 5.1, provide an optional means of grouping messages by conversation in a folder. More primitive clients can achieve a similar effect by sorting on the Subject field, which often remains unchanged over a conversation.

The bulk of the standard email views in IBM's Lotus Notes r6 show a sorted list of messages typical of other email clients. A "Thread View" mode (somewhat hidden in the user interface) displays email messages organized much like a typical Usenet newsgroup client.

These same Usenet newsgroup and email clients support conversations in another, less-overt way. When the user initiates a reply to a message, its contents are copied to the new message. With the text editor the user may leave these contents unchanged, trim the parent message to select a salient portion, or sprinkle replies throughout the quoted content. This operation, called *quoting*, to some extent puts the reply into its conversational context [14]. If the quoting in a series of replies upon replies is allowed to accumulate, it represents all the ancestor messages, i.e. the path to root in the conversation tree.

Prior Research

Several research teams have developed visualizations and user interfaces for representing email and Usenet newsgroup conversation trees.

Loom [3] provides a view of messages in a Usenet newsgroup where each message is a dot placed on a 2D grid: the horizontal axis is time, and the vertical axis represents different authors. Lines connect the dots of a message to its replies. Different conversations are connected with lines of different colors. A message dot can be opened into a separate window to show the message header and contents.

Conversation Map [13] shows the conversations within a Usenet newsgroup using an array of radial tree thumbnails ("spider webs"). A conversation tree can be opened into a separate window to show a larger view of the radial tree, a list of participants and other elements. A particular message can be opened into a separate window.

ConverSpace [11] visualizes a conversation by laying out the message bodies on a 2D grid: the horizontal axis represents time, and the vertical axis represents "topical structure". Laying out the actual message contents into a tree structure is unique in the prior literature.

Netscan [16] shows a Usenet newsgroup conversation as a tree where individual messages are small glyphs displayed in a unique mix of chronological and sequential order. The timeline of the conversation is broken into days; messages on the same day and in the same branch are presented sequentially, but the chronological relationships among messages in different branches in the same day are not represented. Selecting a message in the tree makes its contents and header appear in a separate pane in the same window.

The prototype client interface by Rohall et al. [12] "combines a traditional list of email messages with a time-based message tree." Selecting a message shows the conversation tree that contains it. Messages are shown in the tree as squares with lines joining a message to its replies. The selected message is highlighted in the tree by showing a thumbnail of its content. The message list entries for messages in the same conversation are lightly highlighted. The correspondence of the messages between the tree and the message list may be surmised by the order, but is not directly apparent. The authors do not state where the actual message contents are shown – presumably in a separate window or a pane in the same window.

Threaded Chat? Threaded IM?

Email and Usenet newsgroups support threaded conversations at their core: each message includes a reference to the one it's a reply to. Online chat and instant messaging (IM) systems, with no such support, present messages in sequence. People have threaded conversations in chat and IM despite the problems it introduces. Harris states that cases where adjacent statements in a conversation are unrelated "are the rule rather than the exception" in chat-like communication systems [5]. She elaborates, "Multiple threads may become entangled, and individual threads are rarely free of disruption by irrelevant messages."

Voida et al. found the same effect in IM, describing branching conversations "extremely commonplace" [17]. Voida attributes the prevalence of branching IM conversations to the nature of IM: in the gray area between synchronous and asynchronous communication. Branching

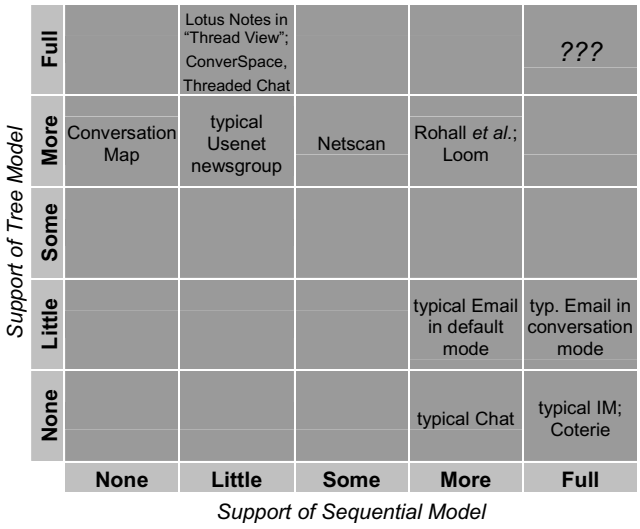

Figure 1: Email systems can be placed approximately in the space defined by the degree to which they support the sequential and tree models of communication. Note that the top-right corner is empty – none of these systems fully supports both models simultaneously.

may arise in part to fill the lulls in the conversational flow while the other person is typing.

Smith et al. designed a threaded chat system to address this problem (and others) [15]. The main element in the user interface to the threaded chat client is a tree of messages much like the one used in Usenet newsgroup browsers. To post a reply in this system the user first identifies the message that is being responded to.

Rather than making the sender specify the message that is being replied to, the Coterie chat client [2] uses heuristics to determine the relationship between messages. Related messages are grouped within a sequential stream; multiple streams can run in parallel within a chat room.

TWO MODELS OF CONVERSATION

Implicit in the designs discussed above are two models of conversation that appear to be in conflict. On one hand, a conversation is a simple sequence of turns; on the other, a conversation is a branching tree.

Sequential Model of Conversation

A visualization supporting the *sequential model* can answer certain questions about a conversation:

A. Which of these two messages was sent first?
B. Which messages were sent before this one?
C. Which messages were sent after this one?

These questions can arise when reading a conversational thread. Question A occurs when the priority of two messages is in question. Question B allows the reader to reconstruct the conversational context of the message's author. Question C directs the reader to all the messages that may contain further discussion about the topic brought up in a particular message. It also arises when returning to a conversation that has received new replies, allowing all the new items to be identified.

An interface supports the sequential model to the extent that it can answer these questions at a glance. Note that displaying the message "sent date" on a non-chronological list of messages does not satisfy the "at a glance" requirement as reading and comparing dates is substantial cognitive act.

Typical email clients in a view sorted or grouped by conversation or subject, typical IM clients and Coterie [2] show a chronological list of messages clustered by conversation, thus strongly supporting the sequential model. Typical chat clients, typical email clients in normal operation, the interface proposed by Rohall et al. [12] and the Loom thread view [3] are chronological but messages are not clustered by conversation, undermining the sequential model because messages in the conversation are more likely to be scrolled out of view. Netscan [16] offers an interesting design point, being chronological by day but mixed within a day. Lotus Notes in its "Thread View", ConverSpace [11], Threaded Chat [15] and typical Usenet newsgroup browsers use a schematic tree view where replies to a message are sorted chronologically; such a tree view can answer some of the above questions some of the time but cannot be relied upon to answer all of the questions all of the time. Conversation Map [11] does not represent the chronological sequence of messages.

Tree Model of Conversation

A visualization supporting the *tree model* of conversation can answer these questions about a conversation:

D. Which message is the root of the conversation tree?
E. Which message is this one a reply to?
F. Does this message have any replies?
G. Which messages are replies to this one?

These questions can arise when reading a conversation. Question D helps the reader understand the original motivation for the conversation's existence. Question E lets the reader resolve anaphoric statements, e.g. if a message says "I agree," the reader can tell by glancing at the message's parent what's being agreed to. Question F lets the reader ascertain whether a topic brought up in a message might have been addressed or elaborated upon. Question G directs the reader to the messages most likely to contain further discussion about the topic brought up in a particular message.

An interface supports the tree model to the extent that it can answer these questions at a glance. Lotus Notes in its "Thread View," ConverSpace and Threaded Chat show the messages in a tree view, and thus strongly support the tree model. Netscan, Conversation Map, the work of Rohall et al., the Loom thread view and typical Usenet newsgroup browsers each shows a schematic tree that reflects the selection of a single message to be viewed, but cannot answer questions E or G at a glance; all these interfaces would typically be used with quoting, so question E would be answered by the message content, leaving G unanswered. Typical email clients rely only on quoting, thus they answer only questions D and E and support the

tree model weakly. Coterie and typical chat and IM clients do not support the tree model at all.

Models in Conflict?

The degree to which each model is supported can be taken as two orthogonal axes. The interfaces discussed here may then be placed approximately in the resulting space captured graphically in Figure 1. Note that the top-right corner of the space is empty.

Is it possible for a visualization to fully support the sequential and tree models simultaneously? We set out to design one.

VISUALIZATION DESIGN

In preparation for designing a *mixed-model* conversation visualization that fully supports both the sequential and tree models, we gathered our design requirements. To begin with, it had to answer all seven questions above at a glance. The "at a glance" requirement ruled out dependence on interaction, e.g. selection or mouse-over highlighting (though we knew we might later add those to reinforce the visualization). That same requirement also ruled out a separate message viewing pane: the message content had to be present in the visualization. From examining many conversation trees, we were aware that they tend to be narrow rather than bushy – that is to say that a message is much more likely to get one reply than two or more – so chains of replies-to-replies should be visualized cleanly.

We chose to list the messages in a chronological, vertical list for three reasons. First, a chronological list of messages supports the sequential model trivially. Second, the messages could be reflowed to fit the available width. Third, scrolling (if necessary) would be in one dimension only. We chose to list the messages from old (top) to new (bottom) so it could be read like a script.

But what about supporting the tree model of conversations? The root is always the first chronologically, so Question D is answered, leaving Questions E, F and G open. Specifically we had to design a visualization that showed the reply relationships. We suspected that indenting would be a crucial part of the layout. Knowing the frequency of reply-to-reply chains, we knew that the first reply to a message would have to be at the same indentation level. It wasn't too difficult to surmise from there that the successive children would be indented successively. Indentation alone didn't convey the message-to-reply relationship, so we knew that some mechanism needed to represent the branching of the conversation tree.

We made numerous attempts at depicting branching. Figures 2, 3 and 4 show some of our interim designs. We implemented one of these (Figure 4) and it failed in user testing. Part of the user interface (not shown) that was being tested alongside the visualization was a schematic representation of the conversation tree much like the one at the left of Figure 2. Our subjects were largely successful at understanding the schematic. Motivated by this partial success, we merged the schematic with the indented messages.

The visualization that resulted from merging the two views is shown in Figure 5, callout (2). The messages are in chronological order, root at the top and newest at the bottom. The root message is at the leftmost indentation level. Its first reply (the second message) is at the same indentation level, connected with a heavy vertical line. The first (and only) reply to the second message (the fourth message) is again at the outer-most indentation level, connected with a vertical line. The second reply to the root message is indented. A heavy horizontal line along the bottom of the root message curves into a heavy vertical line down to its second reply. This line passes beneath the intervening message. Heavy circles provide visual connections between the heavy lines and the message bodies.

We believed that this visualization effectively answered all seven questions at a glance, meaning that it supported both the sequential and tree models. This was verified during a usability study, described later.

It's interesting to note the similarity of the design shown in Figure 5 and the preliminary prototype from a year earlier shown in Figure 2. The salient difference between the two is that in the

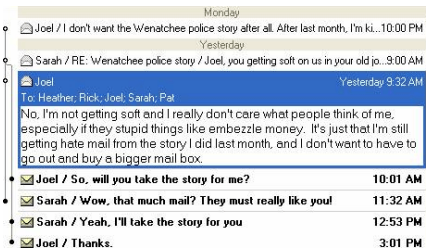

Figure 2: Our first functioning prototype of a conversation visualization showed messages in chronological order with a conversation tree schematic on the left. The selected message is expanded inline; others are collapsed.

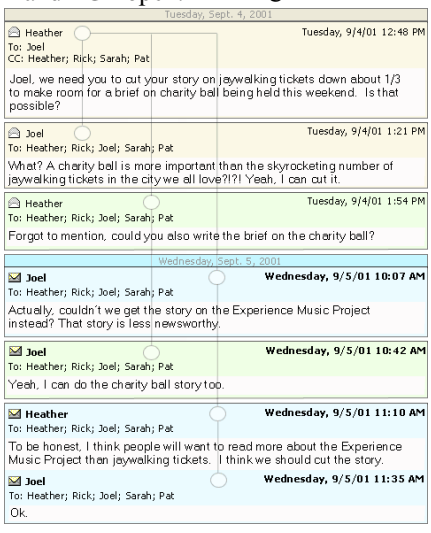

Figure 3: This design sketch showed the conversation tree schematic overlaid atop the message bodies. This was one of many design sketches that were not implemented.

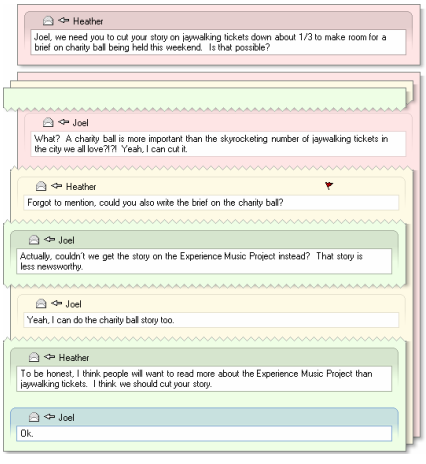

Figure 4: This functioning prototype showed the branches of the conversation tree as layered, colored "slabs". This design did not fare well in user testing.

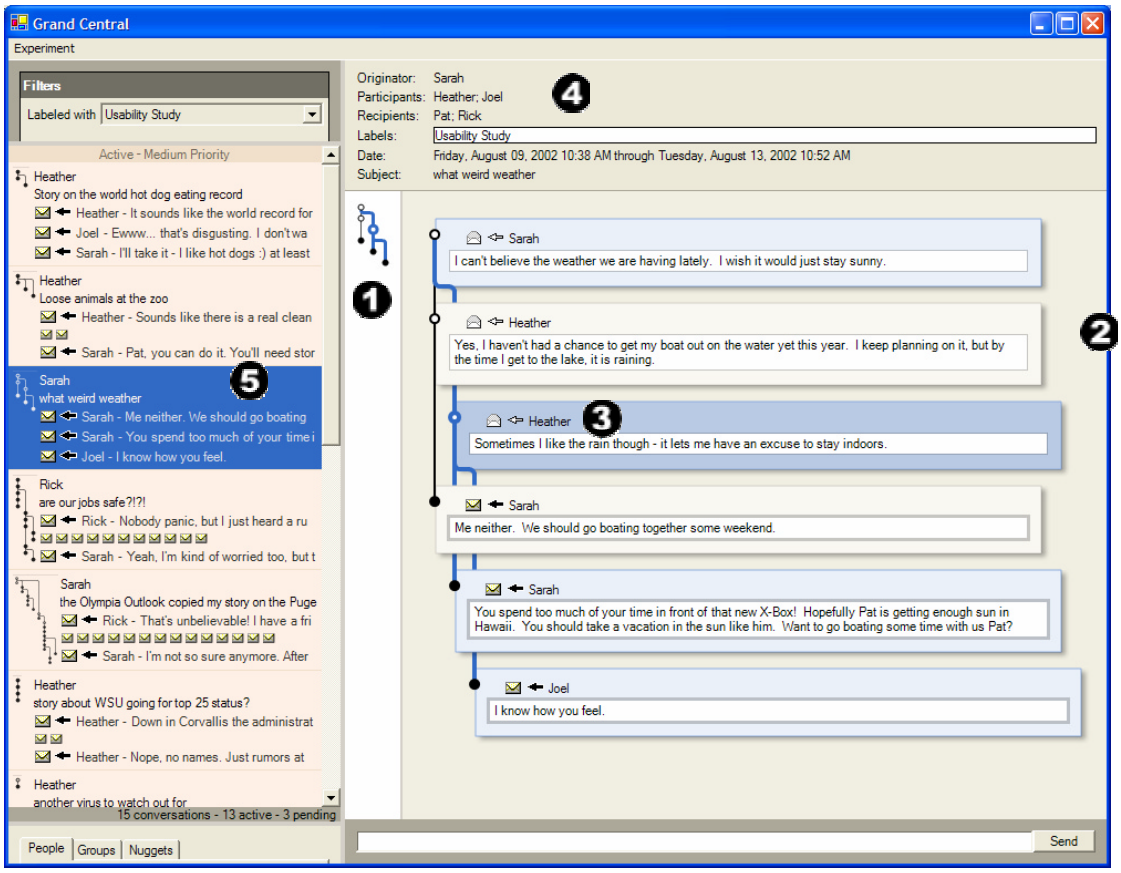

Figure 5: The conversation visualization incorporated into a conversation-based email client. (See text for callouts.)

later visualization the messages are shifted to align with the schematic view of the conversation tree. It's difficult to explain why we, as it were, walked the long way around the block to get next door. Having explored many points in the design space, though, gives us more confidence in the present approach. In the earlier visualization it was difficult to follow the long, parallel, vertical lines. Paradoxically, obscuring parts of those lines appeared to make them easier to follow.

Cleaning the Data

Two things had to be done to make the message data presentable in the visualization. First, quoting actually worked against the visualization by introducing redundant content and inflating message length. Our implementation strips quoting when it is judged to not have changed. Second, messages sometimes appear to be younger than their children. This can happen when system clocks are out of sync. Our implementation repairs any out-of-order times.

Layout Algorithm

Our design renders a conversation tree by first assigning each email in the conversation with a row and column position. Rows are assigned in chronological sequence where the earliest email in a conversation is assigned row zero, the next email is assigned row one, and so on. Columns are assigned based on a conversation's tree structure using a recursive algorithm, given here as pseudo-code:

```
int MaxColByRow[0..MaxRow-1]
function int AssignColumn(Object message, int col, int parentRow)
    for int i = parentRow+1 to message.Row
        col = Max(col, MaxColByRow[i]+1)

    int childCol = col
    bool firstChild = true
    for each child in message.Children
        childCol = AssignColumn(child, childCol, message.Row)
        if firstChild then
            col = childCol

        childCol = childCol+1
        firstChild = false

    message.Column = col
    for int i = parentRow+1 to message.Row
        MaxColByRow[i] = col

    return col
```

This algorithm allows for columns to be used by multiple branches in the conversation tree, keeping the width of the tree's rendering to a minimum (see Figure 6). After each email has been assigned a row and column position, rendering is trivial.

PUTTING IT IN CONTEXT

The mixed-model conversation visualization is an important part of a user interface to support a conversation-oriented email client user interface (UI). Additional parts of the client UI (see Figure 5) are described in this section.

Figure 6: A conversation tree schematic showing column reuse (second and third branches) and column positioning (first branch).

Conversations and messages can be arbitrarily large; unfortunately the same cannot be said of our display devices. We designed the visualization to fit within a user-controlled window width, and to scroll vertically as needed. Scrolling is in direct conflict with the requirement that all relationships be visible "at a glance". There are several established ways of helping users keep context while scrolling [8, 7]. We chose to use an "overview+detail" approach, using the schematic as an overview ((1) in Figure 5) to the detailed conversation visualization (2). The schematic is laid out the same as the detailed visualization, but without the message bodies. A gray band indicates the area of the overview that's currently visible in the detailed view (not shown in Figure 5 because this conversation doesn't need scrolling). It may be that the schematic is useful even when scrolling is not necessary.

(The schematic view of the conversation tree is similar to that shown by Rohall et al. [12]. As mentioned earlier, their visualization doesn't answer Questions E and G at a glance. By interleaving the messages with the schematic, the questions are answered.)

Another way that large conversations are accommodated is by automatically reducing the scale of the schematic and the indent of the detailed conversation visualization.

Because there are numerous menu commands that pertain to a particular message, we needed to introduce the ability to select messages in the conversation. This gave us the opportunity to identify the messages that relate to the selected message. The selected message is highlighted in blue, the selected message's parent and children are highlighted in a lighter blue and the ancestors and descendents are highlighted more lightly still. The lines connecting the message to its children and through its ancestors to the root are made heavier and blue, both in the schematic (1) and the detailed visualization (2). Selection can be then used to help focus on particular branches of a conversation tree.

We normally show a minimal message header (3) – little more than the sender's name and an "unread" flag. The header can be expanded to show all message fields.

An area of the screen (4) is devoted to summary information about the conversation. The first three fields show the name of the person who sent the first message in the conversation ("Originator"), other people who have sent messages in the conversation ("Participants") and people and groups who have received but not sent messages ("Recipients"). Other fields show the labels that have been applied to the conversation, the date range spanned by the conversation and the subject of the first message.

The three panes described above, (1), (2) and (4), provide different views of a particular conversation. The conversation that is being viewed is the selected member of the conversation list (5). The entries in the conversation list show a thumbnail-sized schematic, the name of the conversation originator, the subject of the first message in the conversation, and an indication of each unread message in the conversation.

TESTING THE CONVERSATION VISUALIZATION

We designed the mixed-model conversation visualization to support both the sequential and tree models of conversation. To test its success at presenting both of these models, we performed a usability study that was designed to answer the following questions about the visualization:

- *Are subjects able to understand the sequence of messages "at a glance" for an arbitrary conversation?*
- *Are subjects able to understand the tree of messages "at a glance" for an arbitrary conversation?*

Usability Methodology

We recruited 6 participants for our study, 3 females and 3 males. All participants had intermediate to advanced experience with Microsoft Outlook 2000 or XP, and had some experience working with threaded email conversations, e.g. newsgroups, work discussion lists, etc. Participants were also all considered to be knowledge workers from a variety of occupations, and had normal or corrected eyesight. Only one participant was not familiar with the concept of thread trees prior to the study.

During the study, participants used the conversation visualization within an email client designed to support conversations (Figure 5). The client was populated with a set of email conversations generated specifically for the user study. Participants did not have the ability to reply to or send emails within client.

Participants were first given the opportunity to explore the email client and build an initial conceptual model. This usually lasted between 5 and 10 minutes. Once the participant felt they had enough initial exploration, they performed a series of seven on-screen tasks with the client. Each task required the participant to find a particular conversation or email within a conversation and then answer questions about the found item. Questions included general usability questions as well as specific instances of Questions A-G. Following these tasks, participants completed a post-test questionnaire where they rated how easy they felt it was to answer each of the conversation model questions using the visualization.

Next, participants received a short training session where the conversation visualization was explained, and in most cases, this turned out to merely be validation of the participant's conceptual model. This segment normally lasted only a few minutes. Last, participants were shown two paper screenshots of the conversation visualization, each containing a different email thread. The message contents were replaced with nonsense text and selection highlighting was eliminated, so participants relied only the visualization itself. For each screenshot, participants were asked specific instances of Questions A-G twice, with the exception of D, which was only asked once.

User Study Results

After observing the participants during the study, it was clear that our email client had several usability issues. For example, when making each email selectable in the conversation visualization, we had failed to make the circle next to the message body a clickable region. As well, participants had difficulties understanding various representations that were used in the list of conversations ((5) in Figure 5). Through observation it was apparent that none of these usability issues directly interfered with the usage and readability of the conversation visualization.

We measured our design's ability to represent the sequential model by three means: observational, subjective and objective (the latter results are shown in Table 1). By observing each participant and the answers given during each task, it was evident that by the third task all participants understood that emails were sorted chronologically within the visualization and thus they had a solid grasp on the sequential model. When asked in the post-test questionnaire to rate from 1 (most difficult) to 5 (easiest) how easy it was to answer Questions A, B and C, the median response was 4.5 (mean=4.4, s.d.=0.6), indicating that subjects' subjective perception was that the sequence was easily read. When reading the screenshots, the combined participant accuracy for Questions A, B and C was 90% (65 correct, 7 incorrect). When responding to questions about the screenshots, all participants responded within one to two seconds, thus supporting the "at a glance" claim for the sequential model.

The same three measures were used to assess the effectiveness of the visualization's representation of the reply tree. We observed that by the seventh task, four of the six participants were able to understand the visual cues used to depict reply relationships between messages. (It should be noted that questions relating to the tree model were not asked until the fourth task.) The fifth participant was able to understand the tree model during the training session. The sixth participant still did not have a solid grasp of the tree model at the completion of the study; this participant is the one who had not been familiar with the concept of thread trees prior to the study. When asked in the post-test questionnaire to rate how easy it was to answer Questions D-G on the same scale as before, the median response was 5.0 (mean=4.2, s.d.=1.0), indicating that subjects' subjective perception was that the reply tree was very easily read. When reading the screenshots, the cumulative accuracy for these questions was 96% (81 correct, 3 incorrect). All participants except the one not previously familiar with thread trees were able to respond to the questions about the screenshots within one to two seconds, thus supporting the "at a glance" claim for the tree model.

Taken together, the three complementary measures of the mixed-model visualization's effectiveness show that participants found it easy to answer questions about *both* conversation models.

	Subjective (n = 6)		Objective	
	Median	Mean (Std. Dev.)	n	% Correct
Sequential Model	**4.5**	**4.4 (0.6)**	**72**	**90%**
A (which first)	4.5	4.5 (0.6)	24	92%
B (msgs. before)	4.5	4.3 (0.8)	24	92%
C (msgs. after)	4.5	4.5 (0.6)	24	88%
Tree Model	**5.0**	**4.2 (1.0)**	**84**	**96%**
D (root)	5.0	5.0 (0.0)	12	100%
E (parent)	4.5	4.0 (1.3)	24	96%
F (has replies)	4.0	4.0 (0.9)	24	100%
G (find replies)	4.0	3.8 (1.2)	24	92%

Table 1: *Subjective* scores are subjects' assessments of how easy it was to answer each of the questions with the visualization, rated from 1 (most difficult) to 5 (easiest). *Objective* scores show how accurate participants were in identifying relationships in two screenshots regarding the seven conversation questions.

CONCLUSIONS AND FUTURE WORK

We have discussed two models of conversation that are embodied to varying extents in a variety of systems that support turn-based, tree-structured conversation. We have proposed a visualization that strongly supports both models, and have presented evidence that supports this assertion. We presented a user interface that puts the visualization in context as a part of an email client.

A robust conversation visualization like the one described here is an important building block of a modern email client. There are ways that the present visualization could be improved, other attributes of conversation that can be incorporated into the visualization and important unanswered questions about the effect of the visualization on usage.

There are at least two ways that we would like to improve the mixed-model visualization. First, it could be more compact. When a conversation consists of a sequence of brief messages, the actual message content is small compared with the surrounding graphics. A more compact visualization would allow more relationships to be visible at a glance.

Second, the visualization is overkill for simple conversations. We know that the majority of conversations are one or two messages long [6]. Our casual observations indicate that simple, non-branching structures are common even among larger conversations. In these cases the "cost" of the visualization doesn't impart any immediate benefit. (On the other hand incurring the "cost" in all cases may make it easier for the user to understand as more complex conversations are encountered and as a particular conversation transitions from simple to complex.)

The mixed-model visualization shows some significant features of a conversation: its chronological sequence and its reply tree structure. There are other aspects of online conversations that may be important as well. The temporal pacing of a conversation is one such characteristic. Rohall et al. have shown one way of conveying pacing [12]. A

similar approach could be taken in our conversation tree overview or in the detailed visualization itself.

An important aspect of conversation is the people who are conversing (or listening). Our interface shows the names of the originator, participants and recipients of messages in the conversation. Others have extracted social networks from email or Usenet newsgroup messages [13, 10]. Integrating such a visualization for a conversation or across a set of conversations may aid understanding.

The people involved in a conversation can change from message to message. These changes should be made apparent in the visualization. Of particular interest are "side conversations" that involve a subset of the participants in a larger conversation.

The way that conversations are visualized may change the way the system is used. Our casual observations suggest that using the mixed-model interface changes the approach to reading email: rather than reading a series of related messages, a conversation is read as unit. This may have significant impacts on the interface. For example, it may not make as much sense to have an "unread" flag per message as to have a "cursor" that divides the conversation into read and unread sections.

More significantly, the nature of the visualization may change the way conversation happens through it. We are interested in comparing conversations that are generated using interfaces that focus on the sequential model, the tree model, or the present visualization.

In the introduction, we suggested that an email client structured around conversations could provide better local context, greater global context, and conversation operations. This paper presents a visualization that provides better local context. We present a list of conversations in a way that may provide greater global context, but this topic needs more study. In this paper we do not address conversation operations. Further, making conversations prominent in the user interface is not a panacea for all the ills of today's email clients. Much work remains to be done to make an email client that truly addresses user needs.

ACKNOWLEDGMENTS

We thank JJ Cadiz for the simulated email content and Ed Cutrell for help with the user study.

REFERENCES

1. Denning, P. (1982). Electronic Junk. In *CACM* 25(3), ACM Press, 163-165.
2. Donath, J. (2002). A semantic approach to visualizing online conversations. In *CACM* 45(4), ACM Press, 45-49.
3. Donath, J., K. Karahalios and F. Viegas (1999). Visualizing conversations. In *Proc. HICSS-32*, ACM Press and personal communication.
4. Ducheneaut, N. and Bellotti, V. (2001). Email as Habitat. In *Interactions* 8(5), ACM Press, 30-38.
5. Harris, S. (1999). Interactional coherence in CMC. In *JCMC* 4(4), USC.
6. Hewett, J. and C. Teplovs (1999). An analysis of growth patterns in computer conferencing threads. In *Proc. CSCL 1999*, Erlbaum, 232-241.
7. Hornbaek, K. and E. Frøkjær (2001). Reading of electronic documents: The usability of linear, fisheye and overview+detail interfaces. In *Proc. CHI 2001*, ACM Press, 293-300.
8. Leung, Y. and M. Apperley (1994). A review and taxonomy of distortion-oriented presentation techniques. In. *TOCHI* 1(2), ACM Press, 126-160.
9. Mackay, W. (1988). Diversity in the Use of Electronic Mail: A Preliminary Inquiry. In *Transactions on Office Information Systems* 6(4), ACM Press, 380-397.
10. Nardi, B., W. Whittaker, E. Isaacs, M. Creech, J. Johnson and J. Hainsworth (2002). Integrating communication and information through ContactMap. In *CACM* 45(4), ACM Press, 89-95.
11. Popolov, D., M. Callaghan and P. Luker (2000). Conversation space: Visualising multi-threaded conversation. In *Proc. AVI 2000*, ACM Press, 246-249.
12. Rohall, S., D. Gruen, P. Moody and S. Kellerman (2001), Email visualizations to aid communications. In *Late breaking hot topics – Proc. InfoVis 2001*, IEEE, 12-15.
13. Sack, W. (2000). Conversation map: A content-based Usenet newsgroup browser. In *Proc. IUI 2000*, ACM Press, 233-240.
14. Severinson-Eklundh, K. and C. Macdonald. (1994). The use of quoting to preserve context in electronic mail dialogues. In *Transactions on Professional Communication*, 37(4), IEEE, 197-202.
15. Smith, M., J. Cadiz and B. Burkhalter (2000). Conversation trees and threaded chats. In *Proc. CSCW 2000*, ACM Press, 97-105.
16. Smith, M. and A. Fiore (2001). Visualization components for persistent conversations. In *Proc. CHI 2001*, ACM Press, 136-143.
17. Viegas, F. and J. Donath, Chat circles. In *Proc. CHI 1999*, ACM Press, 9-16.
18. Voida, A., W. Newstetter and E. Mynatt (2001). When conventions collide: The tensions of instant messaging attributed. In *Proc. CHI 2002*, ACM Press, 187-194.
19. Vronay, D., M. Smith and S. Drucker (1999). Alternative interfaces for chat. In *Proc. UIST 1999*, ACM Press, 19-26.
20. Whittaker, S. and Sidner, C. (1996). Email overload: Exploring personal information management of email. In *Proc. CHI 1996*, ACM Press, 276-283.
21. Yiu, K., R. Baecker, N. Silver and B. Long (1997). A time-based interface for electronic mail and task management. In *Proc. HCI International 1997* v. 2, Elsevier, 19-22.