# TREETABLES FOR READING ARCHIVED DISCUSSIONS

Paula S. Newman
Palo Alto Research Center (PARC)
3333 Coyote Hill Road
Palo Alto, CA., 94304, USA
+1 650 812 4239
pnewman@parc.xerox.com

## INTRODUCTION

The MailContent project at PARC is concerned with developing aids to the exploration of archived discussion lists and newsgroups and, by extension, on-line forums. One result of the project is the two-dimensional TreeTable visualization [10] for threads.

The sections to follow develop a rationale for the visualization, by: (a) reviewing the role of tree structures in retrospective reading of threads, (b) contrasting the properties of one- and two-dimensional representations for coherent reading, and then (c) discussing some additional, TreeTable-based mechanisms that alleviate the problems of scale of the otherwise very useful two-dimensional forms. A final section discusses an important gap in content-only representations.

## INVOLVEMENT, COHERENCE AND TREES

The coherence of a document might be measured in terms of the ease with which readers can understand how the parts of the document logically relate to each other and to the whole. A distinction is sometimes drawn between local and global coherence [3], where local coherence concerns the logical relationships between small adjacent/nested segments, and global coherence the relationships among larger units.

Adapting the local/global distinction to turn-taking conversations, we might consider a conversation to be locally coherent if the participants can relate each contribution to what has gone before, and globally coherent if they can accurately summarize the content of the discussion. Local coherence can exist even with a lack of what has been called "sequential coherence" [4], that is, a logical connection between a contribution and its immediate predecessor, if sufficient positioning information ("to go back to what Mary said") is given.

Inbox messages associated with a discussion list thread have less sequential coherence than conventional conversations, because responses to different messages are interspersed. Yet this does not always present a problem [4], and, by personal experience, it almost never does for highly involved participants. For example, participants in a list devoted to the development of an international standard will generally read each message posted to the list as it arrives, and understand how it relates to one of several developing conversations. This is not only because of explicit references to, and excerpts from earlier messages, but also because active participants devote a significant amount of time to considering each message, and they are well acquainted with the subject matter, and with the interests and predispositions of the discussion participants.

It might be observed at this point that the role of a message in an inbox-based conversation, as understood by an active participant, might not be the same as that implied by a tree structure derived from standard linking devices (such as "in-reply-to" fields and prefixed or suffixed messages) and even embedded quotes. Because this is a conversation, a message may take into account, to some degree, all prior messages wherever they occur in the formal tree. Nevertheless, the formal structure is often quite accurate, because message authors do generally feel obligated to position their message as a response to the most directly related earlier message.

As we move away from task-oriented, inbox-based reading, to the reading of archived discussion lists and newsgroups, reader involvement decreases along one or more dimensions. Less attention is generally given to each contribution, and many readers may be less familiar with the subject matter and contributors. So a time-based presentation of messages makes heavier demands on readers in this situation, in terms of imposing coherence, and formal tree structures become a preferred mode of presentation. Also, when less time can be devoted to reviewing a conversation, efficiently readable representations become relevant to global coherence.

The formal tree structures that are used as guides for retrospective reading are generally shown in linear, indented form [2][5]. However, this is not necessarily ideal. Below we review the properties of different tree representation alternatives in terms of their ability to facilitate coherent reading.

## TREE REPRESENTATIONS AND NAVIGATION

Figure 1 illustrates four different ways of visualizing trees, using a thread from the rec.motorcycles newsgroup as an example. Figure 1a shows a two-dimensional, classic node + edge representation with one additional property, namely that subtrees are discrete in the sense that they can be enclosed in nested, but not overlapping, rectangles. Figure 1b shows the same tree in the form of a TreeTable [10], a 2D tree representation developed to

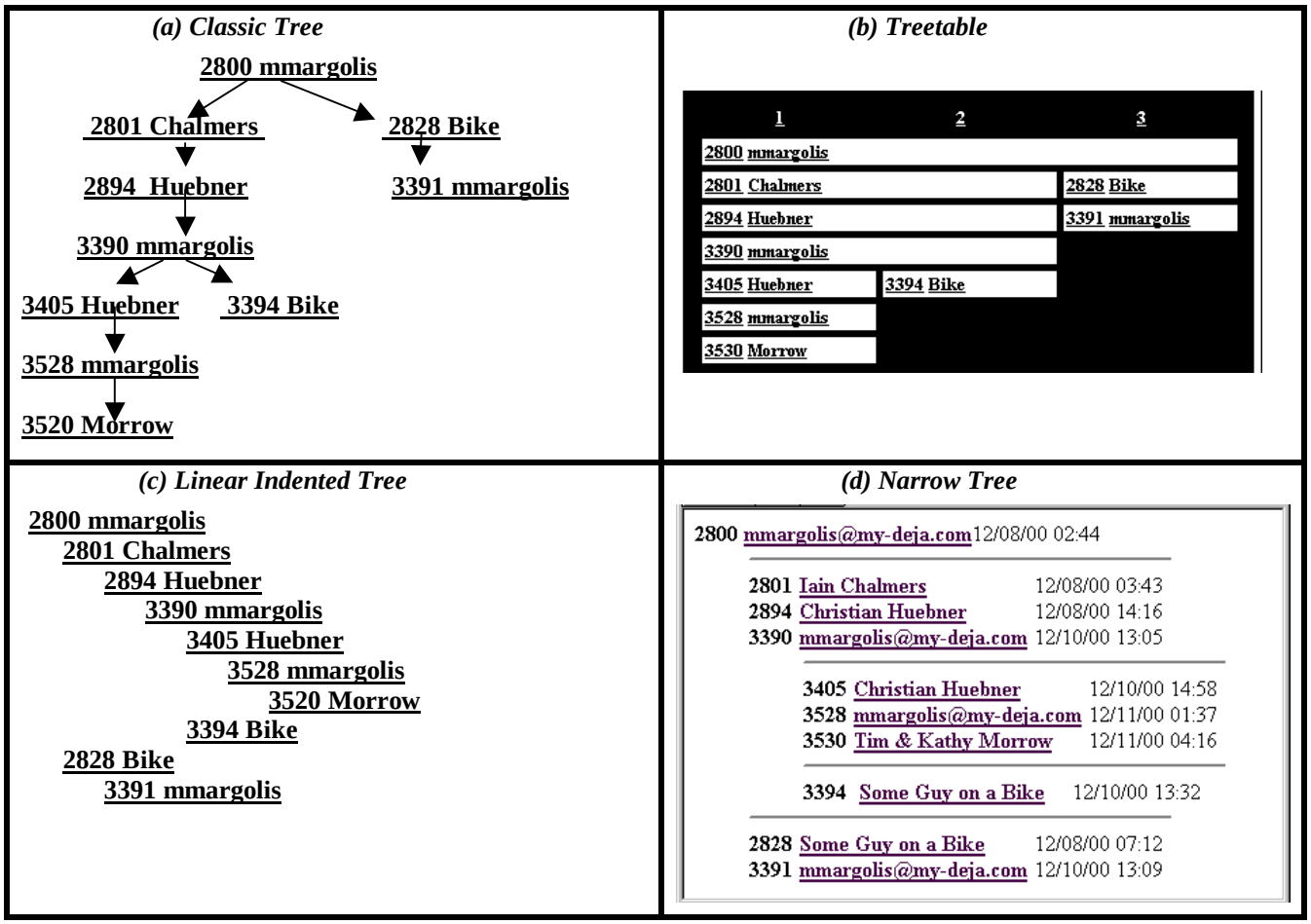


**Figure 1**

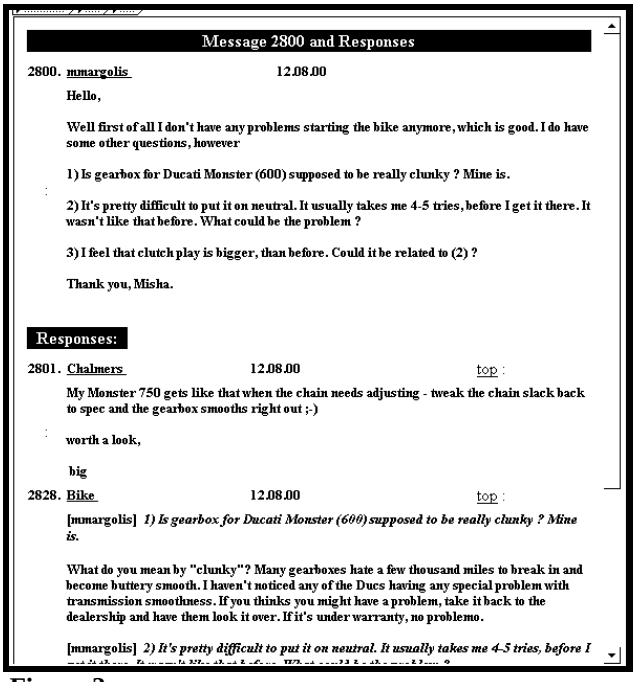


**Figure 2**

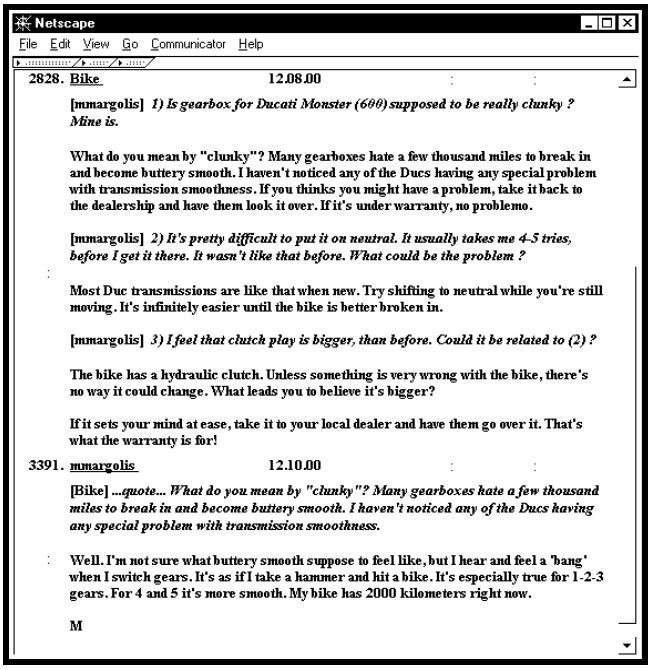


**Figure 3**

avoid wasting space occupied by edges, elucidate paths, and allow text incorporation (discussed further on).[1] Figure 1c is a conventional, one-dimensional, indented tree representation, while Figure 1d is a variant of 1c, called a NarrowTree [10], also developed for text embedding.

Now assume that we want to navigate through the tree message by message, using one of these representations as a guide. One reasonable way to read the content of the thread is to begin by reading the root message, then its immediate responses. The subtree headed by one of those responses might then be selected for deeper exploration, based on the response content and/or the depth of the subtree it heads, etc. Another approach might be to begin by reading all the messages along the longest path, to get a picture of the possibly most substantive part of the conversation.

The two-dimensional representations seem preferable for this kind of navigation, because:

- 2D forms allow locally coherent reading in either breadth- or depth- first order. If the immediate predecessor of a message must be reread, it is easily identified and accessible. In contrast, linear indented trees are convenient only for depth-first reading. And, even in that case, after a message at a leaf of a long subtree is read, it may be difficult to scroll back through the outline to identify and access the predecessor of the next message in sequence. This problem can be reduced by the use of progressive disclosure, opening and closing levels one by one, but doing so further obscures structure.
- 2D forms present a clearer picture of the overall structure of the thread, and of the context of the individual messages being read. This is important for global coherence, that is, for the gradual buildup of an overall model of thread content. Also, if the nodes on a particular level are ordered in time sequence (here they are not) some approximation of the time element is given as well.

The two-dimensional versions also lend themselves to auxiliary displays allowing efficient reading of closely related messages. To facilitate breadth-oriented reading, a listing concatenating all responses to a given message, such as that shown in figure 2, can be provided. The listing can omit prefixed or suffixed messages because they are not needed for coherent reading and, with sufficient message analysis (see [10]) also omit other extraneous material such as signature blocks, yielding a compact, efficiently read form.

The TreeTable version (1b) also lends itself to reading concatenated, reduced, root-to-leaf paths, because these are precisely the columns of the table. Figure 3 shows a listing of column 3 of Figure 1b, anchored at the second message. The result is, in many cases, very much like a quickly assimilable dramatic script.

Nevertheless, linear indented representations are the conventional means of representing threads because they can be extended indefinitely, and thus can portray very large trees within a limited width. The vertical scrolling required is generally not disorienting. In contrast, some of the advantages of the two-dimensional representation are lost for wider trees, because extensive horizontal scrolling *is* disorienting, and thus some nodes must be reduced until they cannot contain identifying text. Some recent approaches using node + edge thread representations (as part of windows containing other material [14] [15]), use tiny nodes incapable of containing any text at all. And an approach that does use text-bearing lattice-like structures for obtaining response inputs [12], illustrates the method with very small threads. We will return to this problem in a later section.

## TREE REPRESENTATIONS AND OVERVIEW TEXTS

Figures 4 and 5 illustrate NarrowTree and TreeTable representations, respectively, of a 54-message thread from rec.motorcycles with embedded initial fragments, and with quotes elided.

Adding some message text to the tree representations can contribute to globally coherent reading by: (a) providing a reasonable idea of the subtopics covered by the thread, and by (b) maintaining context for readers, reminding


**Figure 4**

[1] Another 2D representation, TreeMaps [6], is not discussed here. TreeMaps portray trees by recursively dividing a rectangle representing the root node into "child" rectangles, sized proportionally to the sums of attributes of their contained leaf nodes. While the lack of edges gives some TreeMaps a superficial similarity to TreeTables, they differ in intent and applicability.

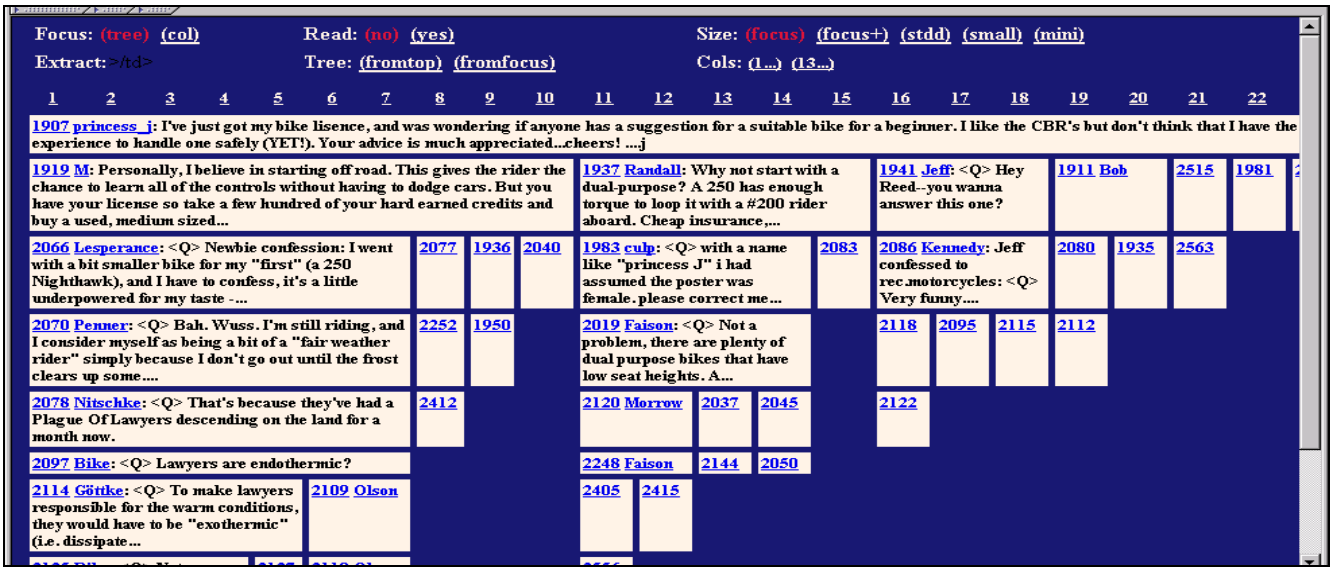


**Figure 5**

them of the issues addressed in the area they are currently reading, areas they have already read, and areas they have not yet read.

The comparative properties of linear vs. two dimensional tree representations are even more starkly evident when text is added. Any amount of nodes and text can be added to indented trees, so that by simple scrolling the reader can gain an overall picture of the subjects dealt with by the thread. However, because the addition of text further obscures the tree structure, this picture is essentially impressionistic, and does not help in maintaining context.

In contrast, while adding text to TreeTables serves both overview and context-maintenance purposes for smaller trees, the utility of the text-embedded forms diminishes as trees grow in width, and more and more cells are too narrow to contain any readable text. In the next section we try to address the problems for treetables posed by larger threads.

## EXPLORING TREETABLES

To retain many of the advantages of treetables for larger threads, we can use focus operations, and also extraction. Figure 6 shows the same treetable as that of figure 5, but with an explicit focus on the subtree headed by message 1937. This approach, an adaptation of fisheye [11] views and, in particular, their use in TableLens [13], allows us to expand subtrees and columns not seen in detail in a neutral view. By expanding the focused-upon elements even further, as in figure 7, many messages can be read in full within the treetable proper. We can also use overview + detail approaches [11] for larger trees, implemented for treetables by extracting subtables into separate windows (not shown).

**However**, none of these approaches are sufficient for exploring very long threads, such as the 160-message thread shown in outline form in figure 8. While the user may extract portions of the thread for study, there is little guidance (except subtree volumes) as to what portions they might extract, or try to read in full. Also, reading such threads in their entirety is simply too laborious.

What is really needed is a thread summary linked to the tree structure, isolating the major subtopics and, for each subtopic, providing some idea of the issues involved. In fact, such summaries would be useful for any threads containing more than about 30 or 40 messages, because (based, to date, only on unsystematic inspection) threads of that size tend to branch into a number of subtopics, so that it becomes more difficult for the non-totally-involved user to grasp the scope of the contained subject matter.

It is beyond the scope of this paper to itemize and discuss the many types of work that might be drawn upon in the areas of topic and topic shift identification, and in summarization, in order to develop thread summaries. However, it should be mentioned that such work requires adaptation to the email thread context. For example, to adapt clustering methods based on word-usage patterns to thread segmentation, it is necessary to determine how to account for quoted words, probably in different ways for different message quoting styles. Similarly, to adapt feature-based summarization [7, 8] methods, email-specific features must be identified. Finally, multi-document summarization [9] seems of major importance, because of the considerable amount of repetition, especially within threads that begin with requests for technical assistance or information.

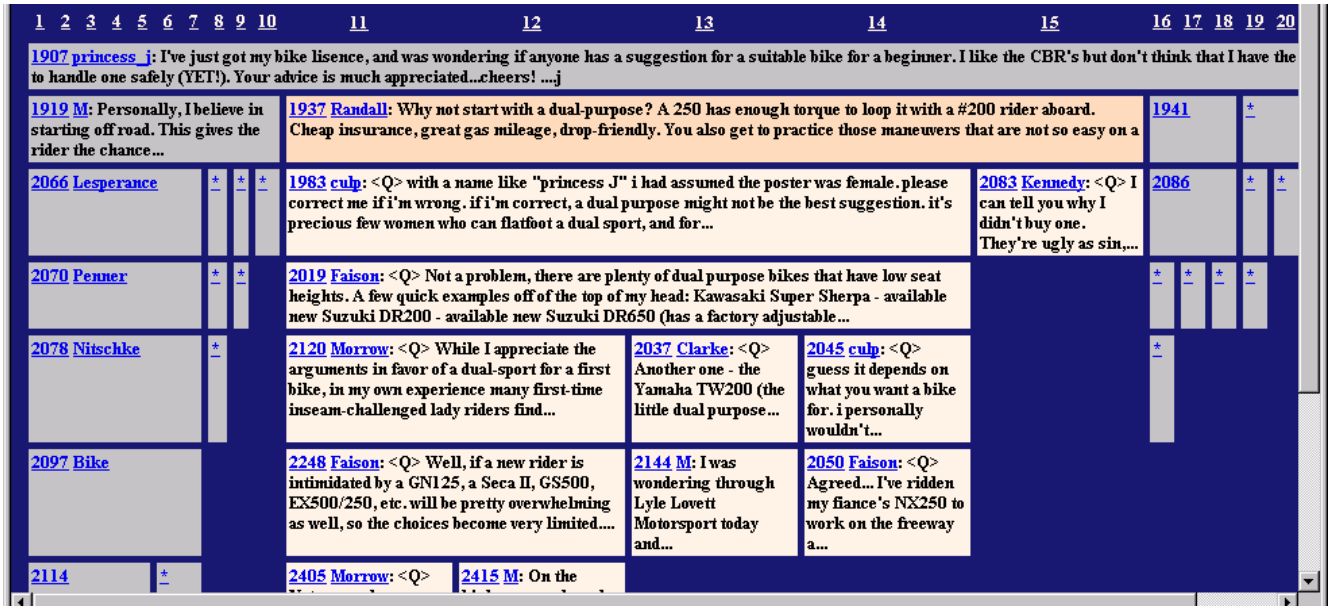


**Figure 6**

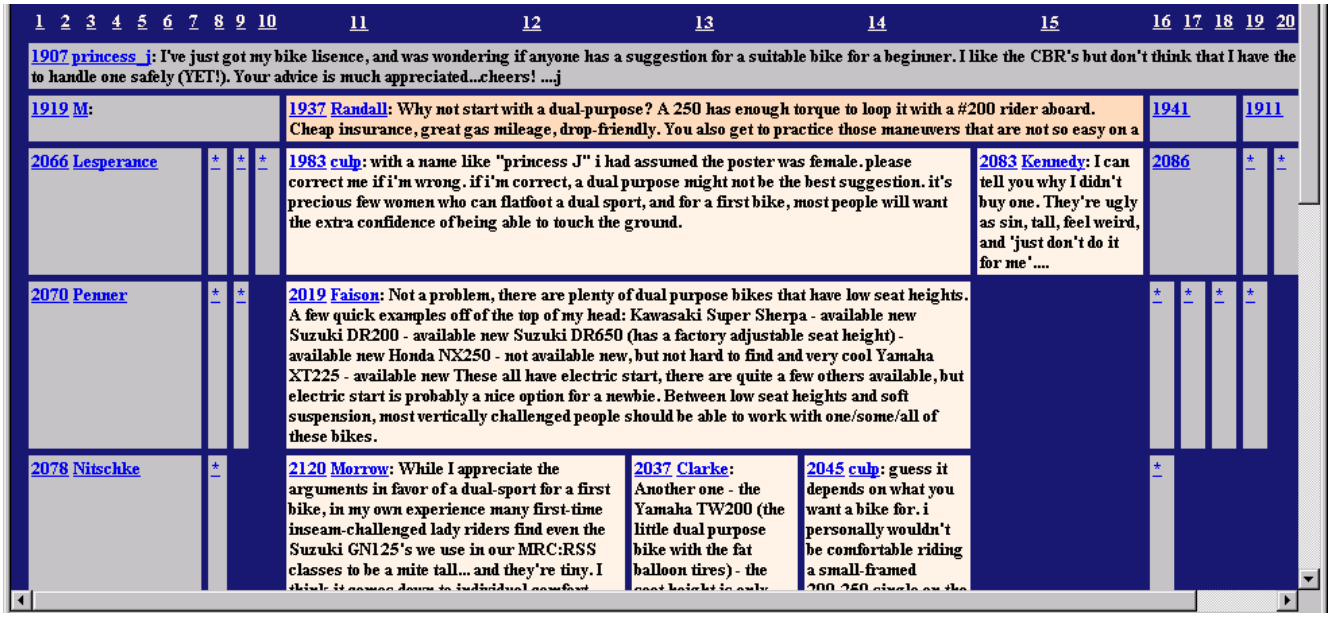


**Figure 7**

## A MISSING DIMENSION

Most of the above discussion has dealt with visualizations of the explicit content of email threads. But, as noted earlier, one of the important cues used by readers to understand a discussion is an understanding of the interests and predispositions of the participants. For example, on task-based lists, some participants may focus on a set of closely related subjects, and/or take a set of closely related positions, sometimes deriving from the interests of an organization they represent. Thus highlighting those relationships contributes a useful dimension to the list discussions.

For this reason, work that addresses the relationships between people and content seems to be a necessary complement to the kind of work discussed here. A pioneering example is Conversation Map [14], which connects sets of people with an approximation of "themes". However, moving this work forward presents major challenges with respect to analysis and approach; either overly precise or inaccurate information might well discourage participation.

## CONCLUDING REMARKS

While linear indented forms have long been used as reading guides for email threads, two-dimensional forms,

in particular, TreeTables, have some distinct potential advantages for coherent, efficient reading. However, further development and experimentation is needed to deal adequately with problems of scale.

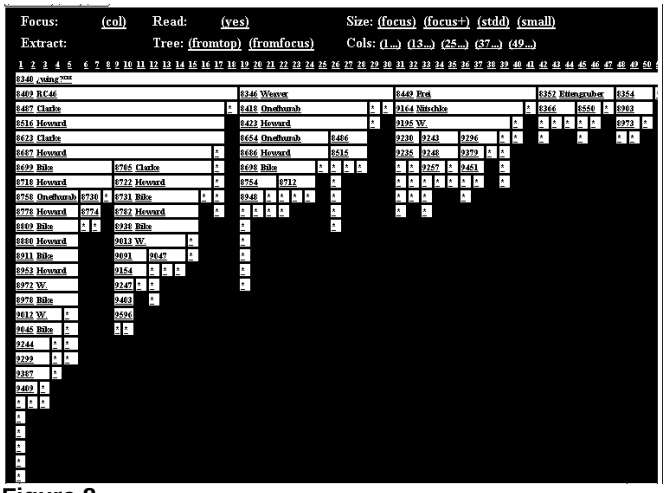


**Figure 8**

## REFERENCES


1. Fellbaum, C., ed., *WordNet: An Electronic Lexical Database*, MIT Press, May 1998
2. Google groups. Available at http://groups.google.com
3. Grosz, B., Joshi, A., Weinstein, S., Centering: A Framework for Modeling the Local Coherence of Discourse, *Computational Linguistics 21*(2) 203-225
4. Herring, S., Interactional Coherence in CMC, *Proceedings of HICSS-32* (Maui, HI, January 1999)
5. Hypermail development archive. Available at http://www.hypermail.org
6. Johnson, B., and Schneiderman. B., Treemaps: a space-filling approach to the visualization of hierarchical information structures. *Proc of 2nd IEEE Visualization Conference* (1991).
7. Kupiec, J., Pedersen, J., and Chen, F., A Trainable Document Summarizer, *Proceedings of SIGIR '95*, (Seattle, WA, July 1995) 68-73
8. Marcu, D., *The Theory and Practice of Discourse Summarization*, MIT Press, Cambridge MA, November 2000
9. McKeown, K., Klavans, J., Hatzivassiloglu, V., Kan, M-Y, Schiffman, B., and Teufel, S., Towards Multidocument Summarization by Reformulation: Progress and Prospects, *Proceedings of AAAI-99* (Orlando, FL, July 1999) 453-460
10. Newman, P., Treetables and Other Visualizations for Email Threads. Xerox PARC Technical Report, September 2001. Available at: http://www.parc.xerox.com/istl/groups/hdi/papers/psn_emailvis01.pdf
11. Plaisant, C., Carr, D, and Schneiderman, B., Image-Browser Taxonomy and Guidelines for Designers, *IEEE Software* 12,2 (March 1995) 21-32
12. Popolov, D., Callaghan, M., and Luker, P., Conversation Space: Visualizing Multi-Threaded Conversation. *Proceedings of AVI 2000* (Palermo, Italy, May 2000) 246-249.
13. Rao, R., and Card, S.K., The Table Lens: Merging Graphical and Symbolic Information in an Interactive Focus + Context Visualization for Tabular Information. *Proceedings of CHI '94* (Boston MA, April 1994) 318-322
14. Sack, W., Conversation Map: A Content-based Newsgroup Browser, *Proceedings of IUI '00* (New Orleans, LA, January 2000)
15. Smith, T., and Fiore, A., Visualization Components for Persistent Conversations. *Proceedings of CHI '01* (Seattle WA, May 2001), ACM Press, 136-143
16. Stent, A., Rhetorical Structure in Dialog, *Proceedings of INLG 2000* (Mitzpe Ramon, Israel, June 2000) (available at link http://www.cs.bgu.ac.il/~nlg2000/final/stent/rst.ps.gz


## APPENDIX: AUTHOR PROFILE (AS REQUESTED)

Paula Newman is a Member of the Research Staff at PARC, where she currently works with an interdisciplinary group investigating facilities for sensemaking over large data collections. She has a long and varied background in research and development, at IBM (in particular at the former Scientific Centers) and then at PARC, at various times focusing on compiler and optimizer development, data base models, and computational linguistics. At IBM she led a number of projects in these areas, and was instrumental in founding and organizing sequences of internal conferences in programming languages and in natural language processing. Her publications in computational linguistics relate to disambiguation, machine translation, and grammar development environments. Her current work on email archives arose out of a task-related review of several large, technically-oriented, discussion list archives, leading to (a) the conviction that the process could be substantially improved, and (b) an interest in the linguistic aspects of this relatively new medium of expression.

To complete the requested profile: Examples of the author's design work are given in the body of this paper. One current direction of interest is the adaptation of Rhetorical Structure Theory to dialog, exemplified by [16], for its potential applicability to thread summarization.