A Tour Through Tapestry

Douglas B. Terry

Computer Science Laboratory
Xerox Palo Alto Research Center
Palo Alto, CA 94304, USA
(415) 812 4427 terry@xerox.com

ABSTRACT

The Information Tapestry is an experimental system that employs both collaborative filtering and content-based filtering, as well as automatic appraising and highlighting, to tailor the delivery and presentation of information to each user's personal interests. This permits users to cope with large volumes of incoming documents, including electronic mail, news wire stories and NetNews articles. This paper presents the facilities available to Tapestry users by way of a scenario depicting a day in the life of an active user.

KEYWORDS

Information filtering, personalized information delivery, collaborative filtering, appraisers, highlighting, electronic mail.

INTRODUCTION

The Information Age, thus far, has resulted in people feeling overloaded with the flood of available information while at the same time feeling insufficiently informed. The Information Tapestry, built at the Xerox Palo Alto Research Center, was designed to solve both of these problems. The system meets its goals by processing information in two steps. First, it filters a large continuous stream of electronic documents into a smaller stream of documents satisfying a user's personal interests. Second, the Tapestry system organizes and prioritizes selected documents such that the user can quickly deal with the high-priority items and read the low-priority, or less interesting, documents as time permits. This paper presents the facilities available to Tapestry users by way of a scenario depicting a day in the life of an active user, namely me.

The next section starts with a description of the information, specifically the electronic mail messages, that awaited me one morning (March 22, 1993). It describes why I got these messages as well as the steps taken to read and process them, including adding new filters to select other interesting messages, saving messages for later reference, and running ad-hoc queries over the Tapestry information database to find related articles. Subsequent sections discuss the status of the Tapestry system, relate some experience with the system to date, and summarize the novel functions supported in the Information Tapestry.

Permission to copy without fee all or part of this material is granted provided that the copies are not made or distributed for commercial advantage, the ACM copyright notice and the title of the publication and its date appear, and notice is given that copying is by permission of the Association for Computing Machinery. To copy otherwise, or to republish, requires a fee and/or specific permission.

COOCS'93 - 11/93/CA, USA
© 1993 ACM 0-89791-627-1/93/0010...$1.50

ONE MORNING'S ACTIVITY

When I arrive at my office in the morning, I am presented with a bounty of new information that has been selected by the Tapestry system. The system has been working through the night searching for interesting articles. I use a single tool for reading this information: an electronic mail reader called "WallTapestry."

Figure 1 shows what the "Active" message folder of my mail reader looks like this morning. It contains over 80 documents (or messages) that have been sent to me for one of three reasons. First, I receive messages sent to me personally by friends and colleagues. Second, I receive messages that have been sent to some distribution list to which I belong. Distribution lists are used heavily within the Xerox Corporation for delivering messages to members of organizations, such as the Computer Science Laboratory with Xerox PARC, or to employees with similar interests, such as those interested in distributed systems. The third reason that I receive messages is because the Tapestry system mails documents to me.

The Tapestry clipping service maintains a database containing articles from the NetNews bulletin board system, the UPI news wire, and other sorts of discussion groups and news feeds. Each day, new articles are added to the database (and old articles deleted). Each Tapestry user has a user profile describing the user's personal interests as a set of queries or *filters* over the information database. Articles that match a user's filter profile are mailed to the user as electronic mail messages. The mail waiting for me this particular morning, as with most mornings, includes 40-50 messages selected by the Tapestry clipping service.

Adobe Inc. v. Pedersen
IPR2023-00986
Exhibit 1017

Adobe Exhibit 1017
Page 01 of 10

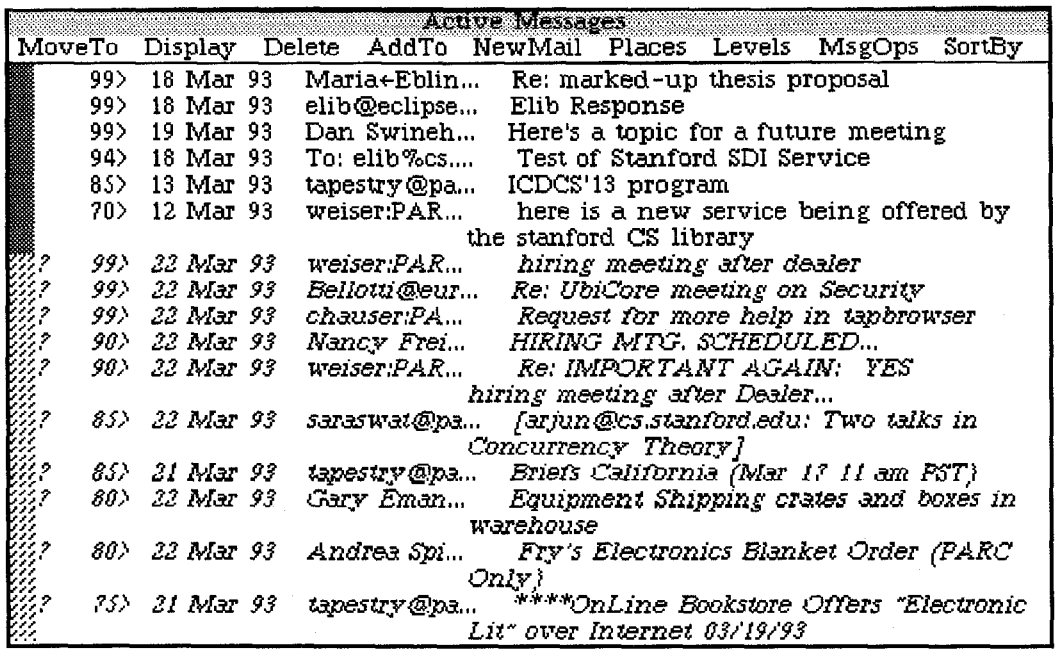

Figure 1: My Active folder showing old messages followed by prioritized new mail.

The messages shown in Figure 1 are presented as the one-line summaries common in most mail reading programs. Each message summary includes the message's date, sender, and subject. Each summary line optionally starts with a "?" to indicate the messages that have not yet been read (these are also in italic font). A unique aspect of the presentation in Figure 1 is the number in each message summary preceding ">". These are priorities that have been assigned to messages by my *appraisers*.

An *appraiser* is a predicate or query that is applied to each new message that I receive. Associated with each appraiser is a priority (and optionally other properties such as a category). If a message matches the appraiser's predicate then the message is assigned the associated priority. You'll note that the messages presented in Figure 1 are not ordered by date as in most mail readers. Rather, the previously read messages come first followed by the new messages. Within the group of old or new messages, the messages are sorted in order of their assigned priority.

The most important messages, the ones with the highest priorities, appear at the top of my mail list so I can read and deal with them first. I often do not get around to reading the lower priority messages until many days later, or sometimes not at all. To me, the highest priority messages are those that have been sent to me personally. One of my most valuable appraisers looks for messages with this characteristic and assigns them priority 99. This morning I have three messages in this category, including one from my boss, Mark Weiser. I read and respond to these immediately, and then either delete them (using the "Delete" button at the top of the window in Figure 1) or move them to another folder for long-term storage (using the "MoveTo" button).

Now I find several messages with priorities in the range 80-90. One of these with priority 85 appears to be a message sent to me by the Tapestry server with the subject "Briefs California." I am curious as to why this particular message was assigned priority 85. So I select the message by left-clicking on its summary, and then I click the "MsgOps" button at the top of the window. This produces a pop-up menu from which I select the "Explain Msg's Priority" option (see Figure 2). The resulting textual explanation is shown in Figure 3.

The message in question was actually matched by several appraisers. One detected that it was sent by Tapestry and gave it a low priority of 10. Another determined that this message was about "Bakersfield" and assigned it a priority of 85. A third appraiser gave it priority 55 because it is a news report, judging from the message's subject. My mail reader uses the highest priority assigned by any appraiser. One could imagine other functions, such as weighted voting, that combine the priorities assigned by various appraisers into a single number. Taking the maximum seems to work quite well in most cases.

I now decide to look at this message by middle-clicking on its one-line summary. This brings up the window shown in Figure 4. While the message has a very long header, only the most interesting (as decided by me) header fields are displayed. One of these header fields, the "X-Filters-Matched" field, was added by the Tapestry clipping service to indicate which of my filters running on the server selected this message. Each of my filters has a unique name so that I can identify them. In this case, the message was selected by my "HometownNews" filter. In other words, I should be interested in this message because it is a news article that mentions my home town (which is a fairly rare occurrence).

Adobe Exhibit 1017
Page 02 of 10

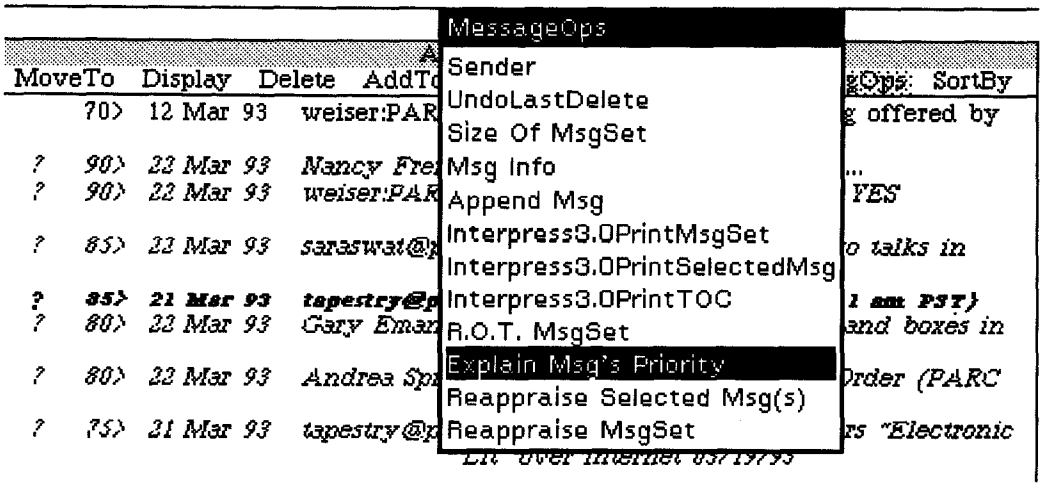

Figure 2: Requesting an explanation for a message's priority.

```
Annotations for message $ XNS-SMTP-Gateway:Parc:Xerox
  appraiser terry$text:Bakersfield => priority 85
  appraiser terry$Subject:Briefs+California => priority 55
  appraiser terry$sender:tapestry => priority 10
```

Figure 3: An explanation of priorities assigned to a message by various appraisers.

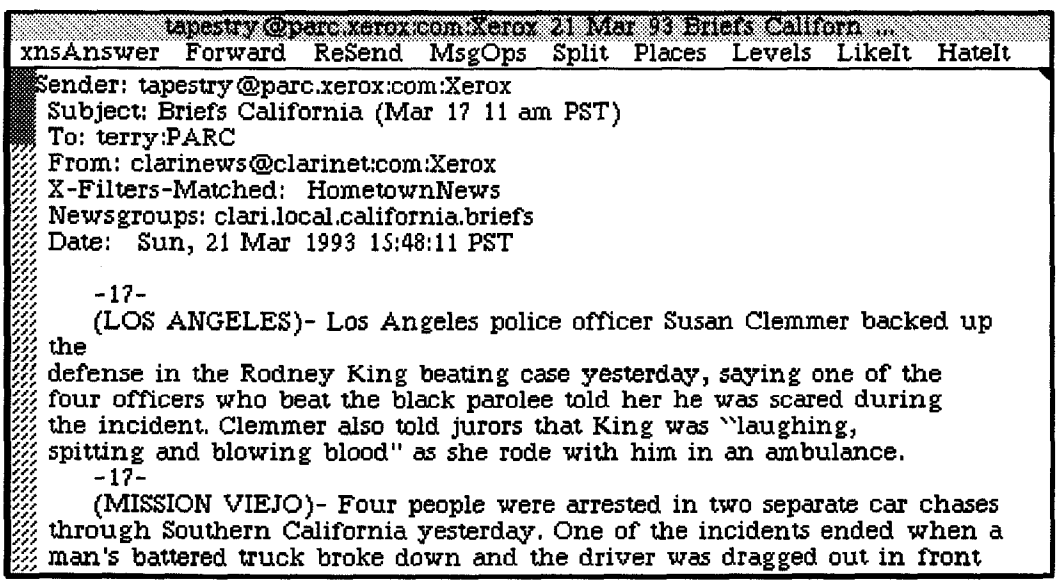

Figure 4: A message selected by my "HometownNews" filter.

The message displayed in Figure 4 is a moderately-long series of short news briefs taken from the UPI newswire and distributed via ClariNet. I know, based on the appraisers that matched this message, that one of the reports mentions my home town, Bakersfield, but I don't know which one. I could scroll through the whole message looking for the report of interest, or I could bring up a search tool and type in my home town's name. Both of these are fairly time-consuming alternatives. Instead, I click on the "MsgOps" menu at the top of the window displaying the message, which once again presents a pop-up menu of operations. I choose the "Highlight Msg's Priority" operation (see Figure 5). This causes the word "Bakersfield" to be highlighted and the message to automatically scroll to the section containing this word (see Figure 6; on a color display, the word's background color would be changed to yellow so that it appears as though the word was marked with a yellow highlighter, but in Figure 6 it is simply shown in reverse video). Well, it turns out that this news story, about Pistacio growers in Bakersfield, is not all that interesting.

Adobe Exhibit 1017
Page 03 of 10

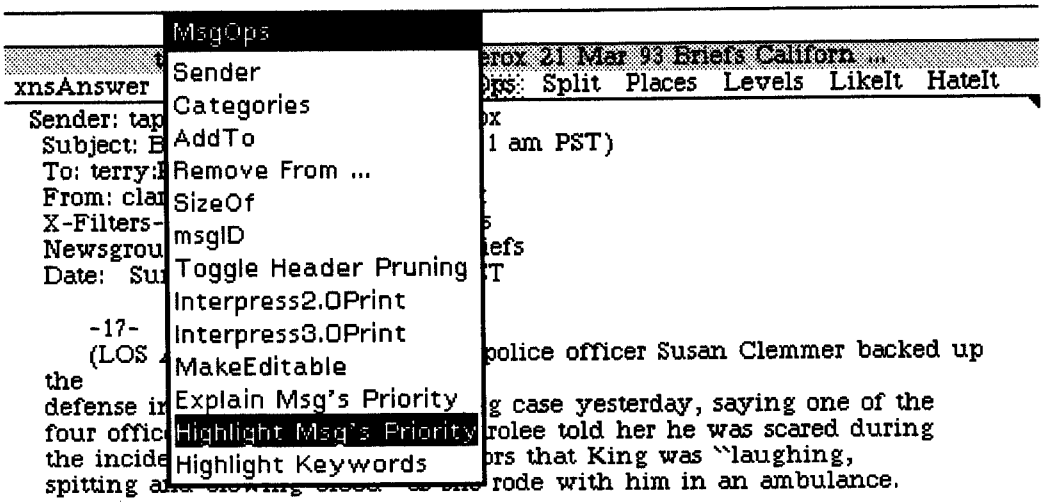

Figure 5: Requesting that interesting portions of a message be highlighted.

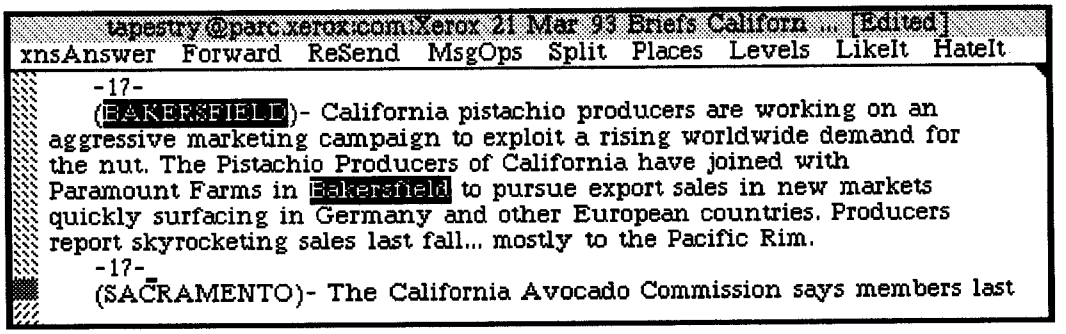

Figure 6: "HometownNews" message with the word "Bakersfield" highlighted.

Further down my list of new mail, I find a message with priority 75 discussing something called the "Online Bookstore" (displayed in Figure 7). Although this message was sent to me by the Tapestry clipping service, it was not selected based on its contents or subject or any other header field of the message. Rather, it was selected by my "SomeLikeIt" filter, an example of *collaborative filtering* in the Tapestry system [3]. It is not uncommon for someone to read a message or article that they find interesting and want to recommend it to others. This particular message, the one shown in Figure 7, was recommended by my fellow Tapestry user, David Nichols. This is indicated by the message's "X-Annotation-likeit" header, which was added by the clipping service at the same time as the "X-Filters-Matched" header.

The Tapestry system permits and encourages users to provide feedback on the messages they read. This feedback is stored in Tapestry's information database and can be accessed in users' filter queries. For example, my "SomeLikeIt" filter query says "show me any articles posted to any newsgroup on any topic that have been recommended by anyone." As the number of Tapestry users increases over time, this filter is starting to select more and more articles. So, in the near future, I will likely replace this overly general filter with more specific filters like "show me articles recommend by David Nichols on the topic of information management" or "show me articles from the comp.os.research newsgroup that have been recommended." The important point is that I decide which articles I see by creating filters based on my particular interests and the people whose judgements I trust.

One of the goals of the Tapestry system was to make it easy for people to recommend articles. My mail reader provides "LikeIt" and "HateIt" buttons that I click to indicate messages that I find particularly interesting or particularly annoying. These are at the top right of each message display window (as in Figure 7 for instance). I probably recommend one or two messages per day on average. Many other Tapestry users read my recommendations. At least one user has a filter that says "show me any articles recommended by Doug Terry."

Continuing to read my mail I find a message with priority "60" that was sent by a colleague (see the message displayed in Figure 8). This colleague read the message on a NetNews newsgroup and forwarded it to various PARC people by pasting it into a message that he addressed to a few local distribution lists. Forwarding the article, as was done is this case, is the traditional method for recommending things.

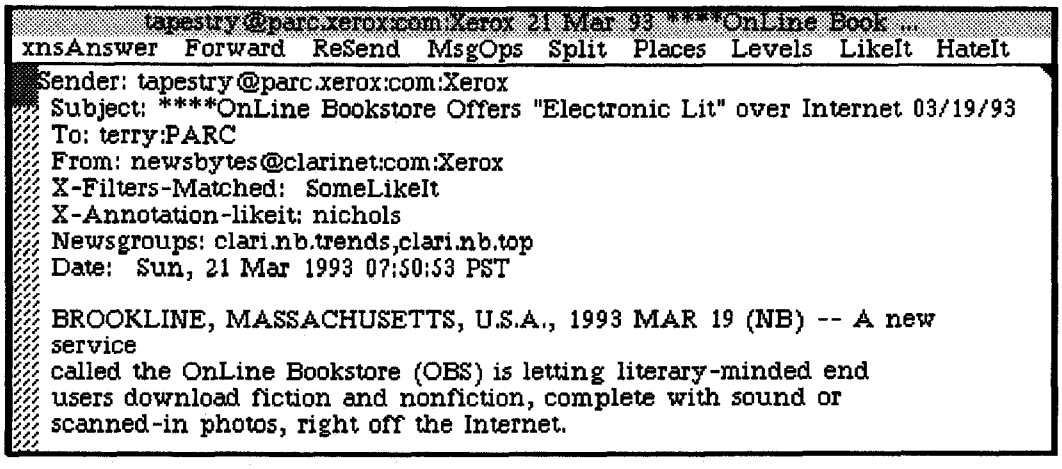

Figure 7: Message recommended by a fellow Tapestry user.

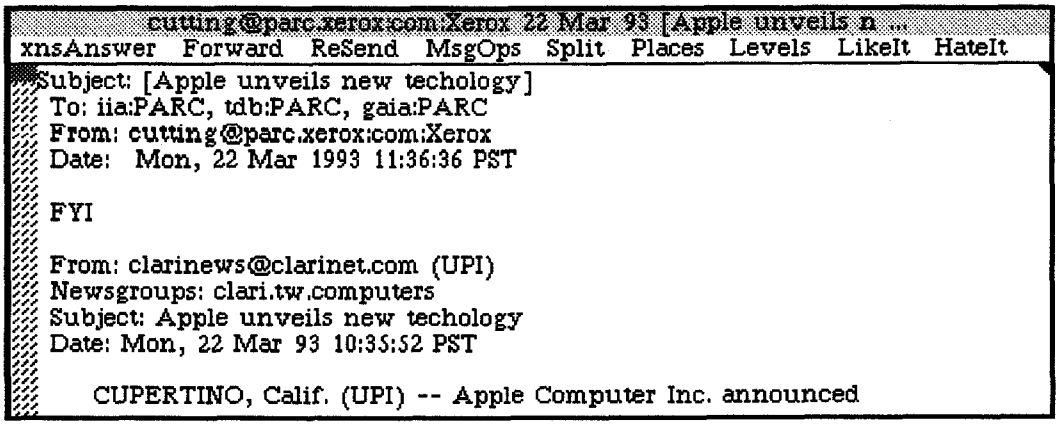

Figure 8: Message forwarded by a colleague.

However, this technique is less than optimal in that it requires a fair amount of work by the forwarder, particularly in deciding who to send the message to, and it forces the recipients to deal with the message, even if they find the message uninteresting. If the colleague were a Tapestry user, then he could have simply clicked "LikeIt". (Of course, the message may have been seen by a different audience depending on who uses Tapestry filters and what these filters select. Even with Tapestry, there are times when one might want to forward a message to ensure that it is received by a particular group of people.)

Reading the forwarded message, I discover that it discusses some new information retrieval technology from Apple Computer called "AppleSearch", something that could very well be relevant to my own work. So, in fact, I am grateful for the recommended message. Moreover, I decide to search for other messages in the Tapestry information database that mentions this technology. To do this, I start up the Tapestry browser (see Figure 9). This tool only queries Tapestry's public information; a separate browser is available for searching my private mail database [4]. Into the top portion of the Tapestry browser I type in the query: m.words = "AppleSearch". This query, written in the Tapestry Query Language (TQL), looks for all articles in the database that contain the word "AppleSearch" anywhere in their body of text. When I click on the "Do It" button, 7 articles are retrieved and summarized in the middle portion of the browser (as shown in Figure 9). Some of these articles, such as the one that was previously forwarded, are from the ClariNet news service, while a few others are part of ongoing discussions on the "comp.mac.hardware" newsgroup. Clicking on one of the summaries displays the article in the lower portion of the browser.

Note that the browser also provides "Like It!" and "Hate It!" buttons (in the middle-to-top right of Figure 9). I click "Like It!" for the article that I had previously seen in order to recommend it to other Tapestry users. With collaborative filtering, I am motivated to recommend interesting articles so that others will recommend articles to me.

I decide that I would like to track articles about "AppleSearch" on a regular basis. Using the Tapestry

Adobe Exhibit 1017
Page 05 of 10

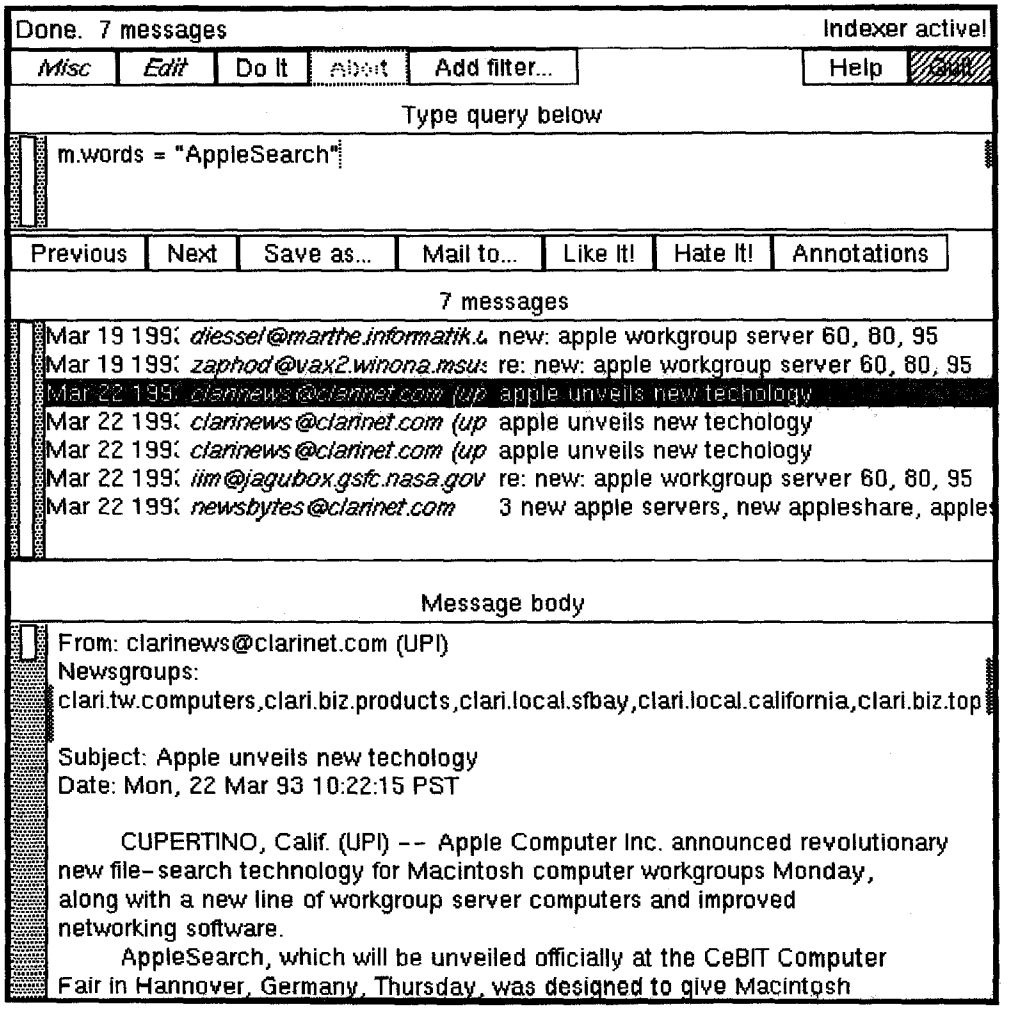

Figure 9: Using the Tapestry browser to find "AppleSearch" messages.

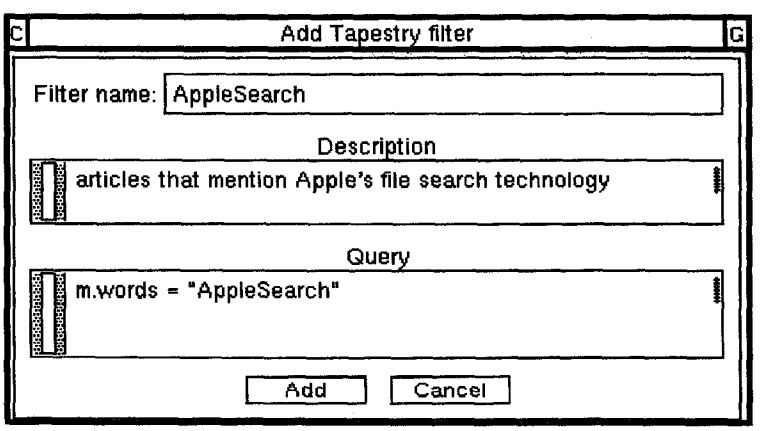

Figure 10: Dialog box for adding a filter from the Tapestry browser.

browser has convinced me that the articles on this topic are informative and the number is not likely to be overwhelming. So, I click on the "Add filter..." button provided by the browser (at the top of Figure 9) to install this query as a regular filter. This produces a dialog box into which I type a name for the filter and a short one-line description (see Figure 10). I also have the option of changing the query. I click "Add", and the new filter is installed. From now on, any message selected by this query will automatically be sent to me.

Adobe Exhibit 1017
Page 06 of 10

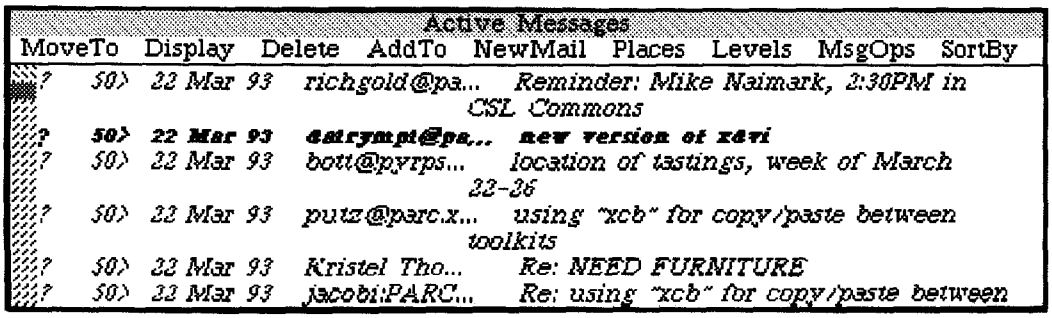

Figure 11: Messages with default priority 50.

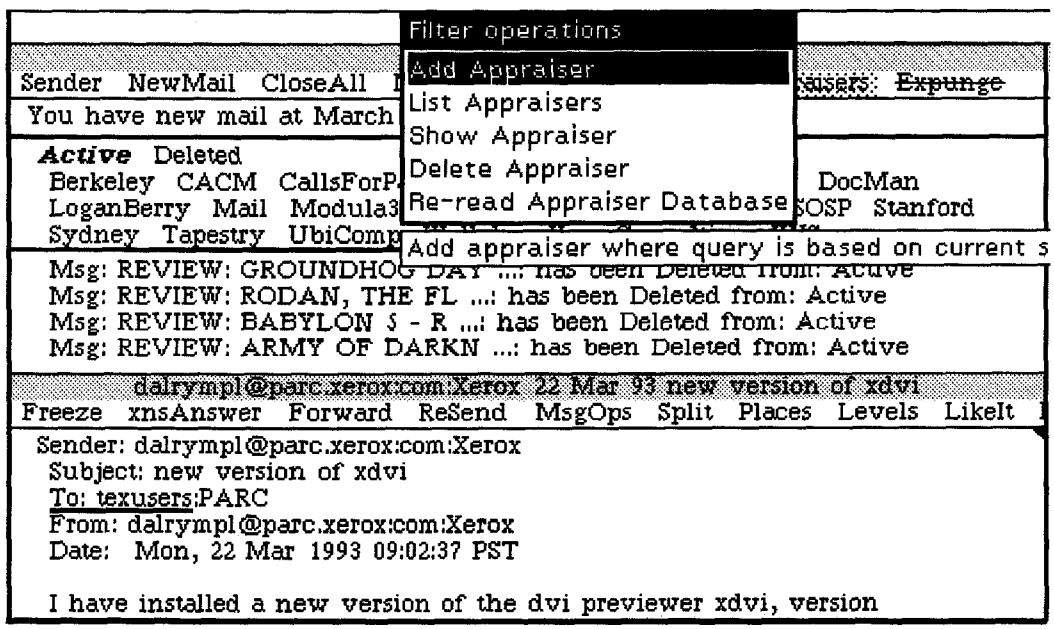

Figure 12: Adding a new appraiser.

I now return to reading the middle priority messages. Several messages have been assigned priority "50" as shown in Figure 11. These are actually messages that were not selected by any of my appraisers. Such messages are given a default priority of 50, a priority in the middle of the range 1-99. Some of my appraisers raise the priority of messages above the default while others lower the priority of selected messages below the default. When I first started using this new mail reader that supports appraisers, most of my mail had the default priority. Over time, I have gradually added appraisers to assign real priorities to the bulk of my messages. Messages that receive the default priority are typically ones sent to a rarely used distribution list to which I belong.

One of the "50" messages I received this morning, the one with the subject "new version of xdvi" in Figure 11, was sent to the "texusers" distribution list announcing a software upgrade. I have switched from using TeX to using a multimedia document editor, so I am no longer interested in TeX announcements. I should really remove myself from this list, but that requires remembering how. I vow to do this later. In the meanwhile, I'll simply add a new appraiser to give a really low priority to TeX messages. To do this, I select the header field of the displayed message that says "To: texusers", and then click on the "Appraisers" button in my mail reader's main control panel. This produces a pop-up menu from which I select "Add Appraiser" (see Figure 12). This, in turn produces another pop-up menu of priorities. I choose priority "5". That should drop any future TeX messages to the very bottom of my message folder.

Although the "Add Appraiser" menu operation only permits me to create simple appraisers, those that do exact pattern matching on individual header fields, my mail reader supports more complicated appraisers. Appraisers may be any boolean function consisting of pattern-matching predicates connected by "AND", "OR", and "NOT." The set of built-in predicate functions include exact match, regular expressions, and soundex matching for user names, and can be extended by registering user-provided procedures that are called on demand. These predicates operate on particular header fields or on the textual body of a message. Appraisers can also assign arbitrary properties to messages, not just pri-

Adobe Exhibit 1017
Page 07 of 10

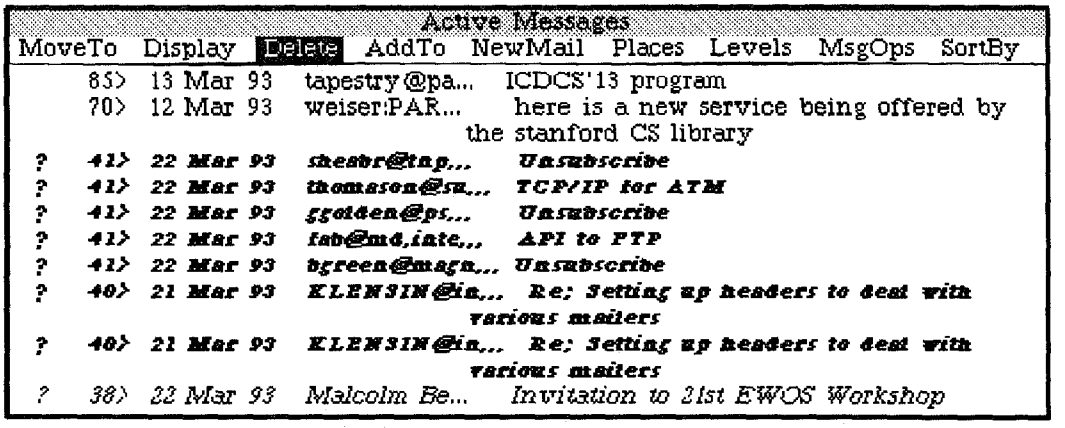

Figure 13: Deleting a block of messages without even reading them.

orities, though a message's priority is the only property currently used by my mail reader. A shell command allows me to add these complex appraisers.

Now that I have read and deleted or saved my most interesting (or urgent) messages, I am left with the remaining lower priority messages. Several of these, the messages with priorities "41" and "40" were sent to the "TCP-IP" internet distribution list. This list has a high volume of traffic that is mostly of marginal interest. Today, I am feeling rushed so I simply select the whole group of messages as a block (this is easy to do since they occur contiguously in my Active message folder as illustrated in Figure 13), and I click "Delete" to get rid of them without even reading them.

You might ask why I even remain on this "noisy" list. There are two answers to this question. First, a small portion of the messages are, in fact. on topics that interest me or are sent by people who usually have something interesting to say. I have appraisers that select these messages and give them higher priority. Therefore, I know that what is left in the 40-41 priority range are the "dregs." Since messages from this distribution list are added to Tapestry's database, I could remove myself altogether from the list and simply receive the most interesting articles via the clipping service. In this case, my appraisers would be replaced with equivalent filters.

The second reason that I remain on some distribution lists is that some days I have more free time than others. On these days I may decide to read some of the less interesting traffic just to see what's happening. The nice thing about having my mail automatically appraised and sorted by priority is that I can decide on any given day "where to draw the line", that is, when to stop reading messages.

As I said before, today I am feeling rushed, as is too often the case, so I quickly dispose of my remaining low priority messages. Many of these I delete without reading them. Some that have interesting subjects I quickly glance at. Others that look interesting, but are not urgent, I squirrel away for another day.

DISCUSSION AND STATUS

The scenario just presented illustrates the various facilities available to Tapestry users. Although this scenario is specific to the new messages that I received on this particular day (included with the permission of the correspondents involved), it is not atypical of the actions I take while reading my mail in the morning. Thus, it also provides insights into how the Tapestry facilities are used in practice. A more scientific study of the overall Tapestry user community, such as that done for the Information Lens [5], remains to be undertaken.

Reading and processing 80 or more messages, as in this scenario, might take me between 30 and 60 minutes on a typical day. Before the existence of Tapestry's appraisers and highlighting capabilities, the time to handle this much mail was at least twice this long. Other Tapestry users have reported similar improvements in their mail reading productivity.

The Tapestry system consists of five main components: the database, clipping service, browser, shell, and mail reader. The Tapestry information database is stored in a relational database management system purchased from Sybase. Over 5000 new messages per day are added to this database, and an equal number of the oldest messages deleted. The current database, containing approximately three weeks worth of articles from NetNews, the UPI newswire, and other information feeds, occupies over a GByte of storage space. The Tapestry clipping service, implemented in the Modula-3 programming language, runs filter queries continuously over this changing database and mails the results to users. The Tapestry browser for running ad-hoc queries is also written in Modula-3. The Tapestry shell, which was not discussed in Section 2, provides commands for manipulating a user's set of filters, including commands for adding, deleting, modifying, and listing filters. Finally, there's the WallTapestry mail reader, which is written in Cedar and runs in the Cedar experimental programming environment. All of the figures in this paper are actual screen snapshots of either WallTapestry or the Tapestry browser in use.

Adobe Exhibit 1017
Page 08 of 10

The Tapestry clipping service and WallTapestry mail reader are separate tools that were built to explore different aspects of recipient-based message delivery [8]. As of March, the time of this writing, the Tapestry clipping service had been in use by its small group of designers and implementors for well over a year. In the past 6 months, the user community has gradually grown to over 30 people within PARC and other Xerox organizations. Because the clipping service distributes results via regular electronic mail, users are free to stick with their favorite mail reader, and most users have chosen to do so. Consequently, WallTapestry is used by only a handful of clipping service users. Similarly, many researchers at PARC use WallTapestry as their mail reader of choice, without using the clipping service. These people either switched to WallTapestry so they could run appraisers over their flood of regular electronic mail, or else were familiar with the Walnut mail reader [4] from which WallTapestry evolved.

The intent of this paper is to demonstrate what it is like to be a user of both the Tapestry clipping service and WallTapestry mail reader and to show how these tools work together to provide a personalized information system. Thus, it focuses on the Tapestry functionality and interfaces, rather than on the algorithms and implementation techniques underlying these facilities. Additional details on the Tapestry architecture, query language (TQL), support for collaborative filtering, and the efficient execution of continuous queries can be found in previously published papers [3][9].

CONCLUSIONS

The Information Tapestry was designed and built to help users avoid information overload while keeping them abreast of information that is important to their daily work, technological development or even personal hobbies. The key to success is making the system aware of a user's individual interests and needs so that it can provide the right information in the right order to satisfy those needs. The Tapestry system is based on three key pieces of technology that serve to successively refine a stream of information:

Filtering selects articles based on per-user interest profiles. Information filtering has been an active area of study for some time and has been incorporated into several systems [1][2][6][7]. The main technical innovation in the Tapestry system is the exploration of collaborative filtering [3]. This permits user participation in the filtering process. Users are willing to supply evaluations of the articles they read if this can be done with little effort, such as one or two mouse clicks, and there is a perceived benefit to the user community. Initial experience with the Tapestry clipping service indicates that filtering information based on user-supplied evaluations yields dramatic improvements over text-based filtering alone.

Appraising automatically prioritizes and categorizes messages. Appraisers, similar to the rule-based agents popularized in the Information Lens [6], analyze each message received by a user. While filters help a user identify *what* to read, appraisers help the user decide *when* to read the selected messages. Or, to put it another way, filters select messages that are potentially of interest to a user, while appraisers further quantify how interesting each message is likely to be. Appraisers can do a more in-depth analysis of each message since they examine a much smaller set of messages than do filters. Like filters, a user's set of appraisers is strongly a function of his or her personal values. For example, a user might specify that "Messages sent directly to me are of top priority" or "Messages on distributed systems are more interesting than talk announcements."

Highlighting indicates portions of a message that are interesting. Highlighting one or more header fields provides users with a visual answer to questions like "Why did I get this message?" or "What is interesting about this message?" For instance, the "from" field of a message might be highlighted to indicate that the message is from the boss. Highlighting words or phrase within the body of a message allows the user to quickly see the passages of interest. It provides a visual answer to questions like "Where does it discuss distributed systems?" This is particularly valuable for long messages that are often a collection of shorter discussions on various topics.

These three functions are essential to information-based systems, such as the Information Tapestry. The user interfaces discussed in this paper are one, but certainly not the only, way of packaging this functionality and presenting it to users. Experience gained through continued use of the Tapestry system will likely suggest additional tools to aid in the filtering and management of large information streams, as well as more effective user interfaces to these tools.

ACKNOWLEDGMENTS

David Nichols implemented the Tapestry browser and a significant portion of the clipping service. David Goldberg, David Nichols, and Brian Oki actively participated in the overall design of the Tapestry system, which also benefited from the ideas and constructive criticism of a great number of my colleagues at PARC. Jim Donahue, Jack Kent, and Willie-Sue Orr were principal designers and implementors of the Walnut mail reader on which WallTapestry was based. Finally, I thank the current Tapestry users who are providing invaluable feedback on the system.

Adobe Exhibit 1017
Page 09 of 10

REFERENCES

[1] ACM. Special issue on information filtering. *Communications of the ACM* 35, 12 (December 1992).

[2] Gifford, D.K., Baldwin, R.W., Berlin, S.T., and Lucassen, J.M. An architecture for large scale information systems. In *Proceedings Tenth Symposium on Operating Systems Principles* (Orcas Island, Washington, December 1985), pp. 161-170.

[3] Goldberg, D., Nichols, D., Oki, B., and Terry, D. Using collaborative filtering to weave an Information Tapestry. *Communications of the ACM* 35, 12 (December 1992), pp. 61-70.

[4] Kent, J., Terry, D., and Orr, W.-S. Browsing electronic mail: Experiences interfacing a mail system to a DBMS. In *Proceedings 14th International Conference on Very Large Databases* (VLDB) (Los Angeles, California, August 1988), pp. 112-123.

[5] Mackay, W.E., Malone, T.W., Crowston, K., Rao, R., Rosenblitt, D., and Card, S.K. How do experienced Information Lens users use rules? In *Proceedings ACM Conference on Human Factors in Computing Systems* (CHI'89) (Austin, Texas, April 1989), pp. 211-216.

[6] Malone, T.W., Grant, K.R., Turbak, F.A., Brobst, S.A., and Cohen, M.D. Intelligent information sharing systems. *Communications of the ACM* 30, 5 (May 1987), pp. 390-402.

[7] Pollock, S. A rule-based message filtering system. *ACM Transactions on Office Information Systems* 6, 3 (July 1988), pp. 232-254.

[8] Terry, D.B. 7 steps to a better mail system. In *Message Handling Systems and Application Layer Communication Protocols*, P. Schicker and E. Stefferud (editors), North Holland, 1991, pp. 23-33.

[9] Terry, D., Goldberg, D., Nichols, D., and Oki, B. Continuous queries over append-only databases. In *Proceedings ACM SIGMOD Symposium on the Management of Data* (San Diego, California, June 1992), pp. 321-330.