CLUES: Dynamic Personalized Message Filtering

Matthew Marx * *and Chris Schmandt*
groucho@media.mit.edu, geek@media.mit.edu
MIT Media Laboratory, Speech Research Group
E15-252, 20 Ames St., Cambridge, MA 02139
+1 617-253-5156

ABSTRACT
Workgroups that defy traditional boundaries require successful communication among people whose interests, schedules, and locations may differ and are likely to change rapidly. CLUES is a dynamic personalized message filter that facilitates effective communication by prioritizing voice and text messages using personal information found in an individual's work environment. CLUES infers message timeliness by considering calendar appointments, outgoing messages and phone calls, and by correlating these "clues" via a personal rolodex. Experience shows that CLUES can be especially useful to mobile users with high message traffic who often access their messages over the telephone.

KEYWORDS: Messaging, electronic mail, voice mail, filtering, personal information management.

INTRODUCTION
Effective communication at a distance is becoming an increasingly essential component of work among groups or partnerships. With a more global economy, many organizations are dispersed or have affiliations around the world, with workers in many different time zones. Modern work groups are as likely to be organized by project or the expertise of particular individuals than by physical location. There is also a rise in collaborative relationships spanning organizational boundaries, often manifesting themselves as ad hoc work groups with short life spans or intermittent activity.

At the same time, the traditional notion of the office is being eroded. At work, much time is spent out of the office in meetings. More employees telecommute. We are increasingly mobile, and employ a range of technologies such as cellular phones, email, and pagers to keep in touch.

But as the volume of messages grows, it becomes increasingly difficult to find the important ones in an efficient manner. Especially when accessing messages remotely, it is often impossible to sort through the dozens or hundreds of messages one may receive in a day to find the few that are immediately relevant. Although rule-based mail filtering systems are appearing in products, it is difficult to configure them; further, once rules are created, they do not change to reflect daily or weekly activity.

This paper describes CLUES, a dynamic message filter designed to automatically prioritize both text and voice messages. CLUES uses a number of sources of information about working relationships to infer which messages are relevant—including calendar appointments, logs of outgoing telephone and email messages, caller ID for incoming voice mail, and a personal rolodex. Implemented as a set of Perl scripts which gather clues about the user's interests at regular intervals, CLUES helps to identify important messages based on these time-varying information sources.

RELATED WORK
A variety of schemes exist for optimizing messaging within a workgroup; these vary in terms of the user investment required to reap the promised rewards. This section describes previous approaches—both their strengths and their shortcomings.

Homogenous systems & "sender's burden"
One approach to improving workgroup messaging is to introduce a tool that, if its conventions are followed by all workgroup members, will result in more effective messaging. This was the strategy behind the Information Lens [4], which allowed the sender of a message to add optional fields to a mail message to mark its priority. Similarly, The Coordinator [12] required that a conversational act be assigned to each message to indicate its purpose: whether it was a request, an acknowledgment, a refusal, etc. The success of such a system depends on the

* Currently affiliated with Applied Language Technologies, Cambridge MA, 617.225.0012.

Computer Supported Cooperative Work '96 , Cambridge MA USA
© 1996 ACM 0-89791-765-0/96/11 ..$3.50

willingness of a message sender to invest the extra work of adding such fields—one might term it "sender's burden." This approach can help a user single out important messages.

The "senders' burden" approach has several limitations, however. In addition to requiring an extra investment of effort on the part of the sender, it denies the recipient control over filtering, as when a power-hungry supervisor marks all messages to subordinates as "urgent" regardless of actual importance. Most significantly, in order for any benefit to be realized at all, a workgroup must standardize on a single mail handler (or a single mail-handling protocol) which can parse the extra fields being added to the messages. Investing time and money in standardization may be difficult to justify or entirely impractical, especially when workgroups defy organizational boundaries.

Rule-based, user-driven filtering

Message filtering may have a higher chance of success when each individual is responsible for filtering his or her own messages. Shifting the burden of prioritization from sender to recipient also eliminates the need to standardize on a single messaging program or protocol. Each user writes a set of rules that will categorize incoming messages. For instance, a message from one's spouse may be marked "important," or any message addressed directly to the user (as opposed to a mailing list or alias) may be labeled "personal." Rule-based filtering, popularized in Procmail [11] and Eudora [1], can prove powerful in separating the wheat from the chaff.

This power does not come without cost, however: writing the filtering rules can be complicated and time-consuming. Rules are written as regular expressions, which can be difficult to author but are easier to modify. As Mackay discovered in her study of customizable software [3], users share filter rules, and a new user usually starts by borrowing someone else's rules.

The amount of time required to write and maintain a set of rules discourages users from writing rules, especially when their schedules or interests change frequently. Thus the investment of effort may outweigh the benefit for all but the most fanatical rule-writers.

Learning-based filtering

A learning-based approach to filtering promises to reduce the amount of effort a user must exert in order to reap the benefit of filtering. Maxims [7], an appendage to Eudora, used Memory-Based Reasoning [10] to prioritize and file incoming messages based on the user's history of email use. It tracked which messages the user read and didn't read, which were deleted and which were filed away. On this basis, Maxims prioritized incoming messages accordingly and suggested what to do with a given message (e.g., read, file, or delete).

Although learning-based systems do not require the user to *do* anything, an amount of time and patience must be invested while the system "learns" the user's preferences. The learning-based approach may be slow to adapt, especially in the case of quickly changing user interests. Further, a learning system may be hard-pressed to judge the importance of a message the likes of which it has never seen before.

Long-term vs. short-term filtering needs

A user's static interests may be served adequately by a rule-based or learning system, since a little effort or patience may pay great dividends over time. To prioritize messages from one's supervisor, for instance, it is probably worth writing a rule or waiting for a learning filter to recognize the pattern. For short-term needs, however, such as getting updates to meeting agendas or hearing back from people one has written recently, neither approach is satisfactory. Writing rules (and, equally important, deleting them once they have served their purpose) will require too great an investment of effort, and by the time a learning system catches on, the need may have passed.

Finding an update to a meeting schedule, getting the response to a message you've sent to a conference chair asking if your submission was received, or catching a call back from someone you tried to reach earlier in the day can be crucial to staying in touch and on schedule. Thus the challenge is to anticipate short-term interests and prioritize incoming messages accordingly.

Fortunately, a user's work environment is typically replete with information about his or her interests. It is these information sources which CLUES surveys in developing a timely model of the user's interests. By forming dynamic models of individual filtering needs without requiring extra effort, CLUES facilitates more effective messaging.

ORGANIZATIONAL CONTEXT AND MOTIVATION

CLUES was designed as part of an ongoing series of projects in remote, telephone-based access to the computer desktop. These projects were designed to serve faculty, staff, and student members of several research groups in a university research lab. Such a work environment exhibits many traits of distributed work groups. Faculty members travel frequently, and students keep irregular hours. In addition to the frequent messages asking for or announcing information, a large number of visitors to the lab results in high mail traffic regarding schedules and agendas.

Phoneshell [8] includes access to email, voice mail, calendar, rolodex, lab-wide dial-by-name, and news/weather/traffic data. Accessing email by text-to-speech synthesis is slow and tedious for non-essential messages [6, 13] but has been found to be extremely handy for small numbers of messages and highly mobile users, since it can be called from airports, car phones, hotels, home, roadside pay phones, etc. Phoneshell first incorporated simple static filtering based on regular expression syntax to match fields in the headers of incoming email against the criteria expressed in the rules. Sample categories might be "very important" (one's boss, major sponsors), "important" (grad students, other members of sponsoring organizations), "personal" (addressed specifically to me) and "other." Since Phoneshell presents higher priority messages first, the caller usually does not bother to read the lower priority messages until at a screen.

The rule-based filtering of Phoneshell is invaluable when trying to cope with multiple days' worth of mail, but the rules themselves are a nuisance to author and maintain. A feature was added to Phoneshell that allows the user to mark a message as important while reading it over the phone. The user can specify what makes the message important: sender, subject, or the sender's Internet domain (i.e., so other messages from the same organization will be important). This frees the Phoneshell user from needing to know regular expression syntax, but has several limitations. First, the user must have waded through all the higher priority messages to find the then low priority message to mark it. Second, once a rule is created it can be removed only by using a text editor. Finally, the relations implied by sender, subject, and domain name are too constrained, requiring exact string matches to fire on subsequent mail.

Our next generation mail reader, Chatter [2], used speech recognition instead of touch tones and memory based reasoning instead of the static prioritization rules. Chatter was never widely adopted, in part due to software bugs and in part due to the difficulty of learning a cantankerous user interface with speech recognition; recognition errors are exacerbated by the task of learning what vocabulary can be spoken. That the effectiveness of memory based reasoning could not be verified in actual use to some extent shows its limitations; the user must invest time in using the system before there is enough experience for it to learn. If Chatter had offered more immediate message management relief, it may have gained a larger audience.

MailCall [5] is the current step in this chain. MailCall also uses speech recognition, and in some more powerful ways than Chatter. MailCall summarizes messages and uses a dynamic recognition vocabulary to let the user then ask for messages by the names of their senders (e.g., "Read me the message from Mark Matthews"). For MailCall we developed CLUES, a dynamic rule-based mechanism which uses information available on the workstation to build new rule sets at frequent intervals (hourly). Such data includes logs of communication, both text and voice, one's calendar, and rolodex. The CLUES filtering mechanism was designed to be independent of the speech user interface, and has been incorporated into the original Phoneshell system, which also remains in use.

HOW CLUES FINDS "CLUES"

Capturing short-term interests requires access to time-varying information about the user. Fortunately, a user's work environment contains several sources of timely information. A calendar, for instance, keeps track of appointments and obligations; since it is indexed by date, one can describe an "interest window" within which entries can be considered timely. A record of outgoing phone calls or email messages lists people the user has tried to communicate with and thus provides clues to user interests. The following sections summarize how CLUES uses these information sources to pinpoint timely user interests.

Extracting clues from the calendar

Establishing user interests with the calendar involves selecting the relevant entries and then extracting meaningful items from those entries. Assuming an "interest window" of approximately two weeks into the future and a few days in the past, CLUES scours individual calendar entries, removing times and useless "stop words" (e.g., "and," "but, " "meeting," or "the").

	MONDAY	TUESDAY
9	Texas Instruments visit	at AVIOS '96 all day
10	show MailCall	fax: 408-506-9987
11		
12	lunch w/Mark Matthews	
1		keynote
2		
3	confirm airline departure	
4	pack for trip	
5	catch 5:15 train to Logan	banquet starting 5:45

Figure 1: Sample entries in a typical user's calendar.

Two lists of clues are generated: one consisting of all the interesting words in the calendar, and the other consisting only of proper noun phrases. (The user can specify which list to use; proper noun phrases (determined by consecutive capitalization) are the more conservative and thus less prone to generating false hits.) For this calendar, the two lists of rules would be the following:

single words	proper noun phrases
Texas	Texas Instruments
Instruments	MailCall
lunch	Mark Matthews
Mark	Logan
Matthews	AVIOS
confirm	
airline	
departure	
pack	
trip	
train	
keynote	
banquet	

Figure 2: Clues generated from calendar entries.

The clues generated from the calendar are then used to find timely messages. For instance, the following message would be marked as timely.

From prender@media.mit.edu Jan 9 15:33:01 1995
To: speech-group@media.mit.edu
From: Rebecca Prendergast <prender@media.mit.edu>
Subject: Final Agenda for **Texas Instruments** visit

CLUES commonly finds messages relating to calendar items, which saves the user the effort of writing the relevant filtering rules because the user is filling out the calendar anyway. (Of course, if the user does not keep a calendar, then CLUES is of limited benefit.)

Finding email replies

Responses to messages one has sent out are usually relevant, and learning of a response quickly allows one to reply in turn and accelerate the asynchronous "conversation." By inspecting the user's log of outgoing email, CLUES can automatically alert the user of responses. Most screen-based mailers offer the option of saving a copy of each outgoing message.

From groucho@media Thu Jan 10 1995 15:35:11
To: **cscw96@acm.org**
Subject: Did you get my paper on time?
...<message body>...
From groucho@media Thu Jan 11 1995 13:12:21
To: msgs@media.mit.edu
Subject: **Anyone know a good mechanic?**
...<message body>...

Figure 3: Log of outgoing email messages.

Determining what constitutes a response is an inexact science, since the form of replies may vary. It is not sufficient to simply detect **Re:** at the beginning of the subject line, for different mailers use different conventions to specify a response. Further, someone may forward a message that was a reply from someone else, which would trigger a false hit. Instead of relying on such tags, CLUES seeks a match for the sender or subject. Both of these messages would be marked timely.

Date: Thu Jan 0 1995 13:45:00
From: **cscw96@acm.org**
Subject: yes, your submission has been received

Date: Thu Jan 0 1995 13:45:00
From: someone@media.mit.edu
Subject: Re: **Anyone know a good mechanic?**

In the first example, the user had written to cscw96@acm.org previously, though the original message had a different subject line. The reply was detected nevertheless since it came from the same address. In the second example, the user had broadcast a message without knowing who might write back. Since the subject lines matched, CLUES inferred that the message was a response.

This strategy for detecting replies is not foolproof, however. Allowing a match on either sender or subject opens the possibility of false hits, as in the case where cscw96@acm.org sends out a broadcast message to all conference registrants. Also, it may fail to detect a reply when someone responds from a different email address and uses a different subject line. But when CLUES does detect replies, communication is accelerated.

Since users may have varying perceptions of what constitutes a "timely" reply, CLUES offers latitude in specifying how far back in the past outgoing messages are to be considered for filtering. Someone with light mail traffic may want to know about a response to anything sent out within the last 30 days, whereas someone dealing with dozens or hundreds of messages per day may want to note replies from only the last couple of days. In a configuration file, the user can specify the number of business days for which outgoing messages are to be considered, and can even specify multiple log files, each with a different time window.

Detecting phone tag

A similar strategy allows CLUES to prioritize responses to phone calls the user has placed (i.e., detect phone tag). When a user places a call using the desktop Phonetool utility [9], Phoneshell, or MailCall, an entry is added to a log file containing the date and the phone number dialed. Since users often look up a name in the rolodex and then dial the number, the person's name and email address may also be available.

Thu Jan 9 1995 13:45:00
(617) 203-9837
Mark Matthews
Fri Jan 10 1995 13:35:12
(415) 265-2466
Pat Featherstone
patf@west.coast.com
Fri Jan 10 1995 04:35:11
(900) 746-7633

Figure 4: log of outgoing phone calls

Filtering voice mail is made possible by a homegrown voice mail system that sends email notification of voice messages, including Caller ID if available. By correlating the phone number in the subject line with entries in the dialing log, CLUES can identify someone returning a call.

> From root@media.mit.edu Thu Jan 9 1995 13:45:00
> To: groucho@media
> Subject: Voice message from x**3-9837**
> From: Operator <root@media.mit.edu>
> You received a 40 second voice message from x3-9837.

Since email messages can be forwarded to a pager, the user can be informed of incoming voice messages without having a separate pager number. And since Caller ID allows detection of returned phone calls, CLUES can help cut down on phone tag.

INFERRING ADDITIONAL CLUES VIA ROLODEX

The clues gathered from the calendar and the logs of outgoing messages and phone calls can become even more useful if they are correlated with additional information found in the user's rolodex. These "second-order" inferences allow CLUES to draw more powerful inferences about which messages are timely.

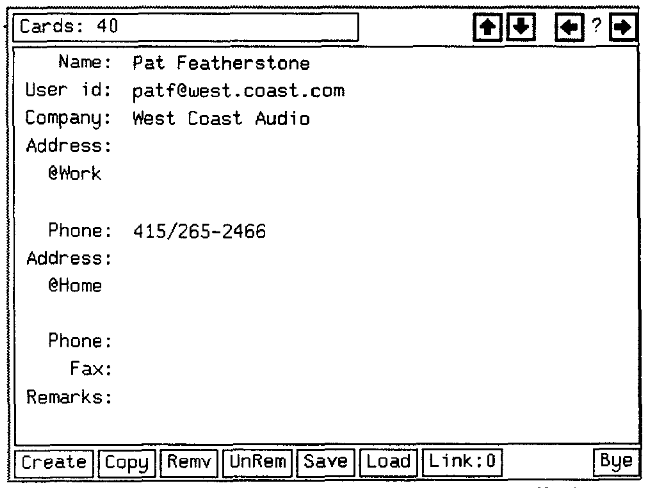

Figure 5: A sample rolodex card. Even if the information is incomplete, certain fields can be useful in finding timely messages.

A typical rolodex card includes the user's name, address, email, and phone. So if CLUES has one piece of information about someone in the rolodex, it can by association retrieve additional clues that may lead to identifying other relevant messages.

Media-independent reply detection

Given the variety of messaging tools available, one often cannot anticipate in what format someone may reply to an outgoing message. People often call in response to an email message or send email in response to voice mail. By cross-correlating the phone numbers collected in the dialing log and the email addresses in the email log with entries in the user's rolodex, CLUES can detect a reply regardless of medium. CLUES looks up a phone number that has been dialed in the rolodex and generates rules for that person's name and email address—and vice versa for an email address. For instance, Figure 4 shows that the user dialed **415/265-2466**. CLUES looks up this number in the rolodex, finding that it matches **Pat Featherstone**. Noting Pat's email address, it infers that messages from **patf@west.coast.com** is important. Thus the following message is marked as timely:

> From **patf@west.coast.com (Pat Featherstone)**
> Subject: I just got your voice mail; here's what I think

Being able to correlate related messages despite differences in media strengthens the claim of voice/text unified messaging for telephone-based message retrieval systems such as Phoneshell and MailCall.

Geographic filtering

The nomad's interests vary not only with time but also by location. Business travelers often leave behind a phone or fax number where they can be reached, and the area code of that number gives clues to where they are and who is geographically close to them—just as a traveler may check an address book to see which friends to visit while on the road. When CLUES detects an area code in a calendar entry, it checks the rolodex for people who share that area code. (CLUES also takes advantage of a knowledge base that represents proximity of area codes to each other. If it finds area code 415, the neighboring 408 is considered, too.)

For example, on the calendar for Tuesday (see Figure 1) is the entry "**fax: 408-506-9987**." CLUES notes that the area code is 408, and so it looks up all the people in area code **408**, **510**, or **415**. It finds Pat Featherstone in **415** and thus infers that **Pat Featherstone** and **patf@west.coast.com** are important, so the following message is marked as timely:

> From **patf@west.coast.com (Pat Featherstone)**
> Subject: Hey, I hear you're in town this week; do you want to get together?

Geographic filtering is especially a convenience for the nomad who pays long distance charges to check messages. Prioritizing messages from people in the city where you're staying can help take care of first things first.

Email domain-based filtering

In addition to finding messages from specific individuals living in an area one is visiting, one may want to pay special attention to messages originating from that person's entire site. So even if I'm going to visit someone at Apple Computer, it's possible that anything coming from Apple will be interesting this week. So if CLUES deduced that Mark Matthews (**mark_matthews@apple.com**) is important since he is in area code **408**, then it will watch for all messages from **apple.com**. So, for example, if one subscribes to a lecture series mailing list for Apple but lives elsewhere in the country, those messages may usually hold only a passing interest. But with an upcoming visit to the area, the user may actually consider attending the talk. The following, for example, will be prioritized:

From lecture-series@**apple.com**
Subject: This week's Friday lecture ......

Not all domains are equally relevant, however. CLUES makes the (weak) assumption that most Internet nodes are geocentric -- that is, that they reside in one location. This is true for many companies and universities (*altech.com* and *mit.edu* are both in Massachusetts). Some companies, however, are geographically distributed: Sun has offices in California and Massachusetts, and IBM is spread all over the globe. CLUES keeps a list of non-geographically specific domains so that it can use the subdomain for filtering (*eng.sun.com* vs. *east.sun.com*)—or, if the location cannot be determined from the subdomain, then it ignores the domain altogether, such as for *aol.com* (America Online). Given the explosive growth of Internet sites since CLUES' introduction, more conservatism in estimating the likelihood of geographic uniqueness may be appropriate.

Here, as in other cases, lies the possibility of false hits. If someone at Apple is on a mailing list to which one subscribes, then suddenly every contribution originating at apple.com will be prioritized—certainly an undesirable result. How CLUES might deal with such shortcomings is discussed in the section on future work.

INTEGRATING CLUES

For CLUES to support filtering in a heterogeneous computing environment, it must both run in a variety of environments and produce output that is usable by a variety of applications. Additionally, it must not impose excessive computational burdens on the user's system.

Setup and configurability

Written in Perl, CLUES runs on most any UNIX machine. (Its effectiveness, however, depends on the availability of data in the user's environment, as described below.) A configuration file specifies CLUES' behavior, including which information sources to use (or not to use) depending on availability and user preference. The configuration file also specifies whether to use single-word rules or proper-noun phrases in generating clues from the calendar.

Since CLUES is designed to model current user interests, it is run as a background process at regular intervals. Experience suggests that running CLUES once an hour is sufficient. CLUES also takes steps to minimize computation, scanning only the information sources that have changed since the last time it was run. It minimizes race conditions generating a shadow set of rules in practice, replacing the master set with an atomic file copy.

CLUES' output

CLUES generates a set of regular-expression rules for various header lines of an email message; several are described below. The first rule finds the proper noun phrase "Mark Matthews" anywhere in the From: line.

```
From:.*Mark Matthews
```

A second, more complicated rule finds the word "Motorola" on the Subject: line, either surrounded by spaces or in its plural or possessive form:

```
^Subject.*[^a-zA-Z0-9]+Motorola((([^a-zA-Z0-9]|es|s|'?s)
+.*)|([^a-zA-Z0-9]|es|s|'?s)*$)).*
```

The next filtering rule, which checks for an email address, will check both the From: line in the header and the local "From " line added by the local mail system.

```
^From( |:.*)friend@high.school.edu
```

The final rule, which matches a subject line, may look unnecessarily complicated, but the various combinations are important to guard against false hits and near misses.

```
^Subject: (Re: )*how about dinner after the AT&T demo?(
-Reply)*\$
```

Although CLUES' output is rather arcane, the user in practice never needs to read or modify the rules.

Application independence

Any application may read the rules generated by CLUES and integrate them into its own filtering scheme. MailCall, for example, supports a user filtering profile with a list of categories and rules for placing messages into those categories, similar to Procmail. Messages that match the rules generated by CLUES are placed into one of the filter categories listed as chosen by the user. Thus both rule-based filtering and dynamic personalized filtering are at work, capturing long-term and short-term interests, respectively.

FILTERING PROFILE FOR USER <GROUCHO>
categories: important:timely:personal:junk
CLUES category: **timely**
User-authored rules: From: geek —> important To: groucho —> personal Subject: cron output —> junk

Figure 6: A personal filtering profile, including categories, a specification for where to place user-authored rules.

CLUES has also been integrated into a visual mailer, HTMaiL, demonstrating that it is independent of medium. HTMaiL reads the user's spool file, categorizes the messages based on CLUES' rules as well as the user's static filtering profile, and writes out an HTML file which can be read by any browser.

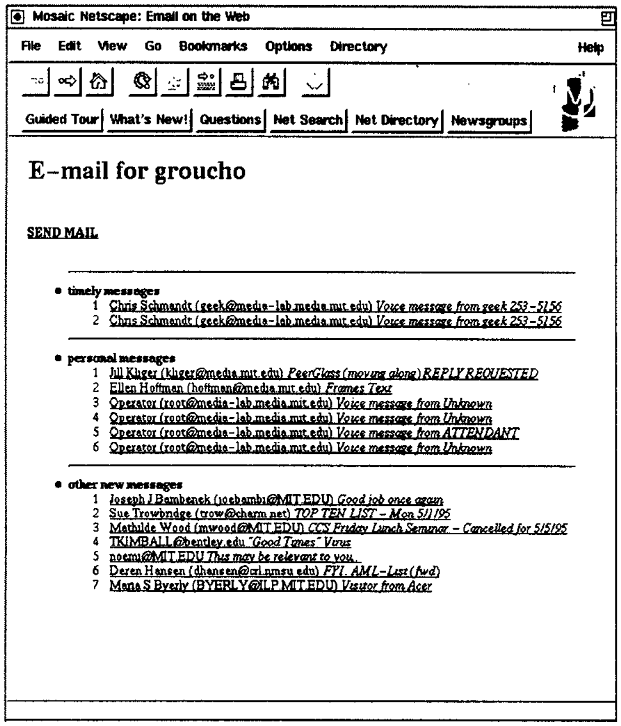

Figure 7: HTMaiL, a visual mailer using CLUES.

One can click on a header line to go to the text of the message. (Copies of messages in the spool file are converted into HTML for easy viewing.) Since the mail spool was parsed into a series of individual messages, this is as simple as making each header line a hyperlink to the file corresponding with that particular email message. For voice messages, HTMaiL not only creates a link to the email notification of the voice mail, but alters the message body such that the user can simply click to hear the voice message. The user may read and respond to messages, but HTMaiL does not support all the functionality of a full-fledged mailer. For instance, no facility is provided for saving messages into files, maintaining aliases, etc. User acceptance of CLUES may motivate the development of a full-fledged mailer which uses CLUES.

INFORMATION INFRASTRUCTURE

CLUES' effectiveness depends on its ability to access a variety of sources of information: call and email log, calendar, and rolodex. That we were able to utilize this information was an unplanned benefit of an open architecture computer/telephony environment, which provided the necessary infrastructure.

Most mail programs provide a means of logging copies of outgoing email, but unless a computer is used to place calls we have no record of telephone activity. We had already developed a screen-based dialer (Phonetool) and an associated rolodex, and had built in mechanisms for any application to access the rolodex database. These tools provide a visual speed dial list, call log, and dialing by clicking on a rolodex card. An ISDN based telephony server also captures caller ID for a workstation-based voice mail system.

Part of the motivation for open architecture is to allow multiple applications access to personal data. Another way to access the rolodex is over the phone via Phoneshell or MailCall; a user can speak or type in a name to initiate a call. Phoneshell also provides calendar access (both read and modify), which insured that the calendar was also available to CLUES.

Much of CLUES' owes to the fact that we experienced immediate payoff after relatively little work, thus motivating further refinements to CLUES. If we had had to first write all the computer telephony applications, CLUES would have been a much larger venture and most likely would not have been undertaken. The good news is that there is currently a strong product trend in computer telephony integration and increasing acceptance of protocols to allow applications such as a calendar manager to publish its contents for use by other applications.

USER EXPERIENCES

We introduced CLUES into our workgroup at the Media Lab in early 1995. (Wider distribution was not attempted since CLUES relies on a particular set of tools described in the previous section; a commercial version, however, would support a wider variety of calendars, rolodexes, and messaging systems.) CLUES was most useful for those with high mail traffic. Those with low traffic tended to read all of their messages when they called in, so hearing the more important ones first, though convenient, was not of great benefit. Those with high mail traffic, namely the two authors, tended only to read the most important messages over the phone, leaving the less important ones until sitting in front of a screen. For them, CLUES was a godsend because it singled out important messages that they might have otherwise not seen until returning to the office. Although one must be wary of claims made by the inventors, the fact that both of us have been using CLUES on a daily basis to filter our messages for well over a year gives some evidence of its usefulness.

One of us travels two or three times a month, accessing his messages almost exclusively by Phoneshell when on the road. Of his 75-100 messages per day, 5-10% are marked "timely" by CLUES. He estimates that for every eight messages CLUES marks as timely, seven turn out to be important messages he would have missed otherwise, and which not have been read until returning to the office. Similarly, if he sits down to read mail and finds that a large number of messages have arrived, he will instead pick up the phone and call Phoneshell to hear his most important messages as prioritized by CLUES, and then return to reading the rest at the terminal.

The other of us is an erstwhile mailing list enthusiast whose mail traffic at one time averaged 200+ messages per day and who had nearly 100 rules in his filtering profile. Although he does not travel often, he too uses MailCall or HTMaiL to find timely messages and deals with the rest later. Even though he works elsewhere now, he still routes most of his mail through the Media Lab to take advantage of CLUES' filtering.

FUTURE WORK

CLUES delivers great benefit despite its simplicity. CLUES uses minimal parsing, no natural language processing, and is generally very simple. What is exciting is that more sophisticated techniques applied to the same problem may yield even more impressive results. This section describes CLUES' current limitations and suggests initial approaches to overcoming them.

Intelligence

CLUES is based on regular expression matching and text-only associations. Although some of its rules can be quite complex, CLUES fails to draw inferences that would strike humans as obvious. For example, CLUES does not know that att.com is the domain for AT&T. It cannot handle nicknames, either, a serious liability. Certain associations, like AT&T-->att.com, could be kept in a table, and a "nickname generator" could take care of nicknames and synonyms, but of course not all cases will be handled.

Perhaps more embarrassing than syntactical oversights are the false inferences CLUES draws since it cannot process semantic information. For instance, if the calendar contains a visit from HP, CLUES will mistakenly mark a message about the "HP printer out of toner" as timely. Incorporating more intelligence and common sense would help CLUES be a far more accurate predictor of timeliness, though we suspect that any such undertaking would require substantial effort.

Feedback

Although CLUES is a dynamic filtering system, it is not adaptive—i.e., it does not improve with experience. Now, the user may adapt, learning to capitalize only the entries in the calendar that are to be used for filtering, for instance. But the user ought to be able to give some feedback about mistaken inferences, and CLUES should incorporate this feedback into future rule generation.

Explanation

When users are surprised by CLUES' choice of timely messages, they ought to be able to ask its reasoning. One can imagine the following scenario in MailCall:

> MailCall: Timely messages. The only one is message 7 from Jim Warner about "Motorola visit 12/1"
>
> User: Why was *that* timely?
>
> MailCall: You have an appointment with Motorola a week from now at 2pm. And you had sent James Warner a message three days ago, and I'm assuming that "Jim Warner" is the same person. Please let me know if I made a mistake.

Such an explanation facility would be especially helpful when accessing messages over the telephone, where one cannot easily inspect the information sources by hand to see why a false inference was made. One reason CLUES does not offer an explanation facility is that it does not consider all possible matches for a message; in the interest of speed, CLUES performs a "short-circuit" search, aborting as soon as it has found a single rule to match a message.

Precision

Finally, CLUES is not very exact. It uses large time windows and rather general notions of timeliness. It doesn't know that an appointment ten minutes from now is probably more relevant than one next week, or that a message from someone you've been emailing and phoning repeatedly for the last four days might be more important than one that happened to match an item three weeks away in your calendar. Whereas CLUES uses a binary notion of importance (i.e., either it's timely or not), weighting the rules by temporal proximity, frequency, or some other measure would be helpful in generating a relevance ranking rather than just a single category of "timely" messages.

CONCLUSIONS

CLUES helps its users work in groups more effectively by allowing them to communicate more easily in a dynamic work environment. CLUES is especially powerful in managing correspondence across organizations, in situations with dynamic outside work relationships, and for the mobile user. Although our experience to date with CLUES has been limited to a small number of users, its unequivocal success for them offers evidence of the power of dynamic personalized message filtering based on personal information available on a computer.

ACKNOWLEDGMENTS

The information infrastructure that supports CLUES' work was developed by numerous members of the MIT Media Lab Speech Research Group. This work was sponsored by Sun Microsystems and Motorola.

REFERENCES

[1] S. Dorner. "Eudora: Bringing the P.O. Where You Live." Qualcomm, Inc. Copyright 1988–1992 University of Illinois Board of Trustees.

[2] E. Ly. "Chatter: A Conversational Telephone Agent" MIT Master's Thesis, Program in Media Arts and Sciences, 1993.

[3] W. Mackay. "Patterns of Customizable Software." *Proceedings of CSCW 90*, pp. 209-221.

[4] T. Malone, R. Grannat, K-Y. Lai, R. Rao, and D. Rosenblitt. "The Information Lens: An Intelligent System for Information Sharing and Coordination." *Technological Support for Work Group Collaboration*, M. Olson (ed), 1989.

[5] M. Marx. "Toward Effective Conversational Messaging." MIT Master's Thesis, Program in Media Arts and Sciences, 1995.

[6] M. Marx & C. Schmandt. "MailCall: Message Presentation and Navigation in a Nonvisual Environment." *To appear in Proceedings of CHI '96*, Vancouver, CA, April 1996.

[7] M. Metral. "A Generic Learning Interface Agent." S.B. thesis, Department of Electrical Engineering and Computer Science, MIT, 1992.

[8] C. Schmandt. "Phoneshell: the Telephone as Computer Terminal" *Proceedings of ACM Multimedia Conference*, August 1993.

[9] C. Schmandt & S. Casner. "Phonetool: Integrating Telephones and Workstations." *IEEE Communications Society, IEEE Global Telecommunications Conference,* November 27-30, 1989.

[10] C. Stanfill and D. Waltz. "Toward Memory-Based Reasoning." *Communications of the ACM*, 29:12 1986.

[11] S. R. Van den Berg. Procmail program. Available from ftp.informatik.rwthaachen.de (137.226.112.172), 1993.

[12] T. Winograd and F. Flores, *Understanding Computers and Cognition: A New Foundation for Design*. New York, Addison-Wesley, 1987.

[13] C. Wolf, L. Koved, and E. Kunzinger, (1995) Ubiquitous Mail: Speech and Graphical User Interfaces to an Integrated Voice/E-Mail Mailbox, Interact '95, 247-252.