COFE: A PROTOTYPE MEMO AND MAIL SYSTEM FOR NON-PROGRAMMERS*

Richard Conway
Steven Worona

TR 82-536
March 1983

Department of Computer Science
Cornell University
Ithaca, NY 14853

*This report supercedes an earlier draft TR 82-489 entitled "COFE: The Cornell Office Environment".

TR-82-536

COFE: A Prototype Memo and Mail System for Non-Programmers*

Richard Conway and Steven Worona
Cornell University

1. Introduction

COFE is a personal filing and mail system intended for non-programmers. It provides a subset of the facilities of conventional editors, file systems and mail systems in a single coherent package with a user interface that is exceptionally simple and consistent.

In addition to this primary goal of offering familiar facilities with unusual simplicity, COFE also provides several unusual features. These include display condensation, annotation, and automatic logging of user actions.

This description is organized as follows:

2. Commands

COFE is controlled by a small set of commands -- **create**, **display**, **send**, **print**, etc. These commands are denoted by boldface type in this description; a complete list of the commands is given in Appendix A.

When special function keys are available, a "soft key" entry scheme is used. Commands are assigned to the function keys, and the assignment is displayed in a prompting line on the screen (see Figure 1). The assignment scheme is described in Appendix A.1.

Alternatively, if special keys are not available, commands are entered textually in a COMMAND field.

Some commands allow (or require) one or more arguments, which are entered in the COMMAND field.

---

* This report supercedes an earlier draft TR 82-489 entitled "COFE: The Cornell Office Environment".

3. Memos

Each COFE user has a private collection of objects called memos. Each memo consists of a body, a header, a history, and optionally, a set of notes:

-The body is a sequence of numbered lines of text. The user can replace any line number with a line-name.

-The header is a group of attributes that identify and locate the memo. The form of the header is the same for every memo.

-The history is an automatic log of actions affecting the particular memo -- see Section 9.

-Notes are supplemental text, that can be attached to individual lines of a memo. In the current version, notes must be entered from the terminal keyboard; future versions will allow notes to be handwritten, scanned or spoken -- see Section 8.

Although all objects in the system are memos, the user can effectively define specialized forms of memos for different purposes (see Section 3.1.1).

All COFE functions are implemented in terms of memos. For example, mail from one user to another becomes a memo in the recipient's library that is essentially a copy of a memo in the sender's library. Similarly, various lists are maintained automatically by the system and others are created by the user. In every case, the list is itself a memo.

3.1. Creation of New Memos

There are a number of way to create a new memo. The most obvious is the **create** command, which creates and displays a new memo, as shown in Figure 1. Other methods of creation are described in Sections 4.1.3, 4.2 and 6.

```
NAME: _________________________________________ LAST USED: 11/15/82
  KEPT IN FOLDER: ________   OF FILE: _________ RETAIN TO: 12/15/82
MEMO NBR: 1075   LINES: 0                       ALERT ON: ________
-------------------------------------------------------------------
       1:
       2:
       3:
       4:
  ...
      14:
-------------------------------------------------------------------
COMMAND:
DESC:
TRAIL:
You have CREATEd a new memo NBR 1075, shown above.
KEYS: 1sk 2sk 3ins 4disp 5next 6list 7send 8accept 9create 10help
```

Figure 1. Screen Appearance after Creation of a New Memo

When a particular memo is displayed, its header is always shown, along with as many lines of the body as can be accommodated on the screen. The body can be scrolled; the header remains at the top of the screen. The header of every memo consists of the following fields:

NBR -- a unique permanent number, assigned by the system.

NAME -- an optional identifier, not necessarily unique, assigned by the user.

FOLDER, FILE -- optional location specifications, assigned by the user.

LINES -- the number of the highest non-blank line in the body of the memo (automatically maintained).

LAST USED -- the date on which the memo was last modified, mailed, printed or **use**d (automatically maintained).

RETAIN TO -- the date after which the memo will be subject to automatic archiving (see Section 11).

ALERT ON -- an optional user-specified date.

3.1.1. Creating a Memo with a Pre-defined Form

The body of a newly **create**d memo is normally empty. Alternatively, the initial body may be copied from some other memo, simply by specifying the source memo as an argument to **create**. Line names as well as line text are copied from the source memo. This means that the user can easily set up a memo to serve as a template, or form, for a special "type" of memo. New instances of this type can be created by giving the name of the template (memo) as an argument to the **create** command.

For example, "**create** appointment form" creates a new memo which is a copy of an existing memo named "appointment form" (assuming there is such a memo).

3.2. Organization of the Memo Library

The user can regard the COFE library as hierarchical: memos can be positioned in folders; folders can be positioned in files. The user can employ these location attributes to classify memos and organize the library. (The location of a memo can subsequently be changed just by altering the FOLDER and/or FILE attributes in the memo header.)

The actual organization of a COFE library is different. Memos are maintained in a list, ordered by their LAST USED date. This list is re-ordered dynamically, as new memos are created, old memos are used, and memos are removed. The early memos on this list form a "working set" of relevant memos. There is a presumption of locality of reference -- the most recently used memos are more likely to be referenced again.

The user sees this chronological ordering as existing within each folder, where in fact, this ordering is the only organization within the library. The FOLDER and FILE attributes are simply descriptors that may be used for retrieval -- two memos with the same FOLDER and FILE values are not necessarily physically close to each other.

This optionally hierarchical organization is convenient but not coercive. The advantages of a hierarchical scheme are available, but the user can alternatively regard FOLDER and FILE as simply second and third names for a memo. However the user employs FOLDER and FILE, the inherent chronological ordering is natural and useful. Moreover, this ordering means that the user is not compelled to give a memo any name at all, nor must a name be unique. Retrieval is still possible, as described in the next section.

4. Retrieving Memos

A central concept in COFE is the selection of a particular memo as the current memo. The current memo is displayed on the screen, and this is the only memo that can be modified, sent to another user, removed, etc. That is, the current memo is the implicit argument to most commands. Consequently, the means by which a memo is retrieved -- made current -- are central to the understanding and use of COFE.

Retrieval is performed by the **display** command. A "description", given as the argument to **display**, specifies a particular collection of memos in the library. There are three types of description:

an integer
: The integer is interpreted as a memo NBR -- the collection consists of the single memo with that NBR, if there is such a memo.

a null description

The collection consists of all memos in the library.

any other description
The collection consists of all memos that satisfy the description.

In every case, the memos in a collection are ordered reverse-chronologically by LAST USED date. The **display** command makes the first (newest) member of the collection the current memo. This is considered the first step of a _display task_.

The task can be continued by means of the **next** command. As successive memos in the collection are made current by the command sequence:

display; next; next; ...

the NBRs of these memos are recorded on the _memo trail_ for that task. For example, the trail might appear:

TRAIL: 662 535 <<942>>

This means that memo NBR 662 was retrieved by the initial **display**, and memos 535 and 942 were retrieved by subsequent **next** commands. The brackets indicate that memo NBR 942 is current.

The **previous** command allows movement to the left along this trail -- that is, it returns to memos previously retrieved in the same display task.

An interesting situation arises due to the dynamic re-ordering of memos during a display task. For example, in the task shown above, suppose memo NBR 535 was "used" while it was current -- say by sending a copy to another user. This use automatically updates the LAST USED date of 535 and makes it the most recently used memo -- hence it should move to the head of the reverse-chronological list. The question is precisely when this occurs. It cannot happen immediately, or **next** would then make 662 current, which would be unreasonable. Hence, re-ordering is deferred until _after identification of the next current memo_ has taken place. But then there is the question of which memo **previous** should yield, if given when 942 is current. The answer is that **previous** is defined with respect to the _trail_ -- it always retraces the memos in the order in which they were retrieved in the current display task, regardless of what re-ordering has been implied by usage during that task. From 942, **previous** re-establishes 535 as current, independent of what action the user may have taken when 535 was previously current in this task (unless it was **remove**d).

At any time, the **display** command without an argument retrieves the most-recently-used memo. This means that in most cases, the user can navigate among the memos of the working set by just a few **display** and **next** commands -- without having to enter any identifier.

4.1. Description of a Collection of Memos

The syntax of a description gives the user great flexibility in specifying a collection of memos. Memos can be described by the attributes the user assigned (NAME, FOLDER, FILE, ALERT date), by automatic attributes (NBR, LAST USED date, RETAIN date), or more generally, just by content. For example, the description

smith IN correspondence

specifies the collection of all memos with the word "smith" in the NAME, that are located in the "correspondence" FOLDER in any FILE in the library. (The case of letters in a description is immaterial. Here we arbitrarily capitalize keywords just to identify them.)

As a second example, the description

smith letter

specifies all memos whose NAME includes the words "smith" and "letter", in any order, to be found in any FOLDER of any FILE. **Display**, with this description as argument, makes the newest such memo current. For example, this might be memo NBR 435 which would be displayed, accompanied by the following command window:

```
COMMAND:
DESC: smith letter
TRAIL: <<435>>
You have retrieved the newest memo fitting that description.
```

If there is more than one such memo, successive **next** commands move progressively through "smith letters" in reverse chronological order, regardless of location. For example, after three **next** commands memo NBR 601 might be current, accompanied by the following command window:

```
COMMAND:
DESC: smith letter
TRAIL: 435 205 436 <<601>>
You have retrieved the next memo fitting that description.
```

From this point, two consecutive **previous** commands would reestablish memo 205 as current, and the command window would appear:

```
COMMAND:
DESC: smith letter
TRAIL: 435 <<205>> 436 601
You have returned to view a memo seen earlier. Note trail.
```

Location attributes are identified in a description by the keywords "IN" and "OF", denoting a FOLDER name and a FILE name, respectively. For example:

smith letter IN personal OF correspondence

smith letter OF correspondence

IN personal OF correspondence

OF correspondence

Descriptions by NAME and location both depend upon _word matching_. A "word", for this purpose, is _any string of non-blank characters_. In the memo NAME, a word is delimited by any of the normal punctuation characters. Neither word order nor case is significant. For example, the description

smith letter

includes memos with NAMEs such as the following:

smith letter
letter smith
letter to John C. Smith
letter to Smith, John
letter: SMITH & SONS

but does not include memos with NAMEs such as the following:

letter to Smithsonian
Smith Co. letterhead
reply from Smith

FOLDER and FILE names are single words (unlike NAME), and location descriptions are necessarily also single words. An exact match (except for case) is required.

An alternative to a word in NAME descriptions is a _string_, indicated by quotes. Blanks are allowed in a string, and any matching pattern satifies. For example, the description

"smith letter"

includes memos with NAMEs such as the following:

SMITH letter
blacksmith lettering

but excludes memos with NAMEs such as:

letter smith
smith, letter

4.1.1. Description by Date

Memos can also be described in terms of the various dates in the header. The significant keywords are ON, BEFORE and AFTER. For example:

```
smith letter ON 3/23/82

smith IN personal BEFORE 10/4/81

IN personal OF correspondence AFTER 7/3
```

Of the three dates in the header -- LAST USED, RETAIN and ALERT -- the default is LAST USED. Alternatively, the RETAIN and ALERT date may be specified:

```
ALERT ON 11/15

RETAIN AFTER 11/20

smith ALERT BEFORE 11/18/82
```

4.1.2. Description by General Content

It is also possible to describe memos by specifying words or strings that can appear _anywhere_ in the memo -- in the body as well as the header. Such words are prefixed in the description with the keyword "WITH". For example:

```
letter WITH smith
```

specifies memos with the word "letter" in the NAME and the word "smith" anywhere in the memo -- in the NAME, FOLDER, or FILE, in one of the date fields, in a line name, or in the text of a line.

4.1.3. Collections in a Remote Library

Normally, a description applies to the user's own private library. However, a user can have access to the "public" FILE in other libraries by adding an AT phrase to the description. For example:

```
smith letter AT wilson
```

The phrase "OF public" is implicit in any description that includes an AT phrase.

Memos from a remote public file are protected against change, nor can they be sent or removed. However, a local copy of a remote memo can be **created**, and the use of the copy is unrestricted.

Special uses of remote access provide a public notice facility (see Section 6.1), and an archival facility (see Section 11).

4.2. Listing Memos in a Collection

An alternative to retrieving memos in a collection one at a time (by repeated **next** commands) is provided by the **list** command, which displays the headers of the next n memos in the collection. (The number n depends on the physical screen size -- it specifies a "screenfull".)

If a new description is given for the **list** command, it initiates a new display task; if no description is given, **list** continues the current display task, listing the next n memos. In general, **next**, **list** and **previous** can be intermixed in any way to control the progress (and direction) of a display task.

The **list** command automatically creates a new memo in which to record its list of memo headers. An example is shown in Figure 2. Except for the manner of its creation, such a "list memo" (it is created in the "lists" FOLDER) is like any other memo in the library. That is, it can be modified, sent, printed, etc., and remains in the library until removed or archived. However, it does enjoy one special privilege with respect to the mail facility -- see Section 6.

```
NAME: smith LIST                                  LAST USED: 11/15/82
  KEPT IN FOLDER: lists    OF FILE:               RETAIN TO: 12/15/82
MEMO NBR: 3024    LINES 6                         ALERT ON:
______________________________________________________________________
       1: NBR 2840  NAME andrew smith
       2:     IN FOLDER resume  OF FILE personnel
       3: NBR 604   NAME smith inquiry
       4:     IN FOLDER     OF FILE sales
       5: NBR 1194  NAME smith
       6:     IN FOLDER personal  OF FILE correspondence
       7:
 ...
      14:
______________________________________________________________________
COMMAND:
DESC: smith
TRAIL: 2840 604 <<1194>>
You have created list memo NBR 3024, listing the next memos.
```

Figure 2. List Memo Created by **list** Command

In general, each **list** command creates a new memo. The only exceptions are when there are no memos to list -- a null list is not created, and when one **list** (without its own description) immediately follows a previous **list** -- the second list is appended to the preceding list memo.

5. Processing the Current Memo

The user may retrieve a memo just to view it, or to process it. This may involve modifying its content, renaming or relocating it, changing the RETAIN or ALERT date, sending a copy to another user, or printing a copy. Only the current memo can be processed, so retrieval is the first step in any processing.

Most computer-based file systems automatically create a temporary copy of an object when it is retrieved, and the user must understand this, and sometimes choose which copy is to be preserved. COFE does just the opposite. Normal processing is directly on the library copy, which automatically reflects any changes made without any explicit "file" or "save" action by the user. When appropriate, the user can easily make a working copy (see Section 3.1.1), but the copy is, in fact, a different memo.

Exactly what constitutes "use" of a memo is important, since this controls update of the LAST USED date and the re-ordering of memos within the library. Modifying a memo, sending it, printing it or copying from it, are obvious forms of use. However, the fact that a memo becomes current does not in itself constitute use -- since the system cannot distinguish between a significant viewing and an incidental or accidental viewing. Consequently, a **use** command is provided, by which the user can force update of the LAST USED date, and thereby bring the memo to the front of the by-use ordering -- in effect, including it in the "working set" of memos.

6. The Mail Facility

The COFE mail facility sends a copy of the current memo to another user. Upon "acceptance" by the recipient, a new memo is automatically created in the recipient's "mail" FILE, with the NAME and body identical to those of the sender's memo. The FOLDER of the new memo is the name of the sender.

The identify and content of mail is preserved by making it impossible for the recipient to alter a memo received as mail in any way except by annotation (see Section 8.1). This means that if such a memo is later forwarded to yet another user, the second recipient can clearly distinguish between the original memo and subsequent additions. (The restriction on the original recipient is not burdensome, since the mail memo can easily be copied and the use of the copy is unrestricted.)

The **send** command, with the recipient's name (or nickname) in the COMMAND field, dispatches a copy of the sender's current memo. Upon arrival, the announcement "New Mail has just arrived" appears in the recipient's message window. Nothing further happens until the recipient gives the **accept** command, which creates the new "mail" memo.

The mail system is monitored by several builtin memos. Each time a memo is sent, an entry is automatically made in the sender's "outgoing mail" memo (NBR 2), listing the name of the recipient, the name of the memo, and the date and time of dispatch. Upon arrival, a similar entry is automatically made in the recipient's "incoming mail" memo (NBR 3). Upon **accept**ance by the recipient, the corresponding entry is automatically deleted from both the recipient's "incoming mail" and the sender's "outgoing mail", and a corresponding entry is made in the sender's "accepted mail" memo (NBR 4).

The timing of **accept**ance is up to the recipient, who also has the option of rejecting mail simply by **delet**ing the corresponding entry from "incoming mail". This deletion also causes the corresponding entry in the sender's "outgoing mail" memo to be deleted, and a corresponding entry to be made in the sender's "rejected mail" memo (NBR 5).

Until mail has been **accept**ed by the recipient, the sender has the option of retrieving it by simply **delet**ing the appropriate entry from the "outgoing mail" memo.

Another builtin memo, "mail addresses" (NBR 6), provides a list of names or nicknames, and the corresponding system addresses. If the **send** argument is found in "mail addresses", the corresponding system address is used for dispatch. If the argument is not found in "mail addresses", COFE next searches for a memo with this argument as NAME. If such a memo is found, it is interpreted as a list of recipients. The first word of each line of this memo is interpreted as the name of a recipient, for whom an address is sought in "mail addresses". A copy of the current memo is sent to each recipient on the list.

If **send** is specified when the current memo is a "list memo" (located in the "lists" FOLDER of any FILE), then this memo is interpreted as a list of memos to be sent to the recipient (or to each recipient, if a list of recipients is given). Each line in the list memo that begins with "NBR integer" specifies a memo to be sent; other lines in the list memo are ignored.

6.1. Public Notices

A user can also "post a public notice" by sending mail to a special COFE user named "notices". This user is special only in the sense that acceptance of received mail is automatic and the memos created are in the "public" FILE rather than the normal "mail" FILE.

Memos sent to notices can be **display**ed by any user by including "AT notices" in a description. Similarly, a local copy can be **create**d.

6.2. Communication outside of COFE

The mail facility does not require that all communicants be COFE users. Gateway utilities accept mail from other systems and dispatch mail to other systems.

7. Interruption and Resumption of Display Tasks

Recall (Section 4) that a display task involves all the memos in the collection specified by the description given with a **display** command. It is often useful to be able to interrupt a task to do other processing, and then be able to resume the task at the point of interruption. There are three different types of processing that might be undertaken while in a middle of a task:

1. processing the current memo

2. performing a "singular task"
3. initiating another display task.

The first two of these do not represent a significant interruption and are described below. The third is described in the following section.

Any action affecting only the current memo is not a significant interruption of the task that made that memo current. That is, one can alter the current memo, **send** or **print** it, and continue the task at any time with **next**, **previous** or **list**. Even if the current memo is **remove**d, the task can still be continued.

A "singular task" involves a single memo, rather than potentially a collection. For example, **display** by NBR is a singular task. The specified memo becomes current, displacing whichever memo was previously current, but no new trail is initiated and **next** and **previous** have no meaning with respect to the singular task. **Create** and **accept** also initiate singular tasks. A singular task is explicitly terminated, and the prior display task resumed by:

resume to reestablish the prior current memo

In addition, a singular task is implicitly terminated by:

next to resume the prior task, and move forward on the trail (backward in terms of LAST USED date)

previous to resume the prior task, and move to the left on the trail

list to resume the prior task, then list the headers of the next n memos in a new list memo.

remove to remove the memo of the singular task, and **resume** the prior task

A singular task is also implicitly terminated by any action that initiates another singular task (**display** by NBR, **create** or **accept**).

7.1. Stacking of Display Tasks

When the user initiates a new display task -- by **display** or **list** with a non-numeric description -- the first memo of the new collection becomes current and the new description and trail appear on the screen. While the user may assume that the prior display task has been aborted, and proceed with the new task as if that were the case, in fact the prior task has only been interrupted. Its description and trail have been automatically recorded in a builtin memo NAMEd "tasks" in the "lists" FOLDER of the "builtin" FILE (memo NBR 1). At any time, the user can return to this prior task by giving the **resume** command. The new display task may itself be interrupted and subsequently resumed in the same way.

The interruption and resumption of tasks is natural and unobtrusive. An unsophisticated user can ignore the facility and behave as if one display task is terminated by initiating another. On the other hand, the facility is useful and convenient for a knowledgeable user. It permits arbitrary excursions into the memo library -- presumably suggested by information being displayed -- without losing the description or position in a prior task.

However, there is a complication involving the chronological re-ordering of memos. The effect of dynamic re-ordering of memos on the progress of a display task was cited in Section 4. The question is, of course, more serious when there are suspended tasks, which depend on an ordering in the library that may no longer exists. The issue is resolved by preserving the complete trail for each suspended task, and causing a **resume**d task to honor the memo ordering in its own trail. That is, a **resume**d task moves backward and then forward over its own trail -- in effect, following the memo order that existed when that task was active. But when a **resume**d task is advanced (by **next** or **list**) beyond the end of the prior trail, it extends the trail along the current chronological ordering of memos. (Note that this means it is possible for a given memo to appear more than once in a particular trail.)

8. Display Condensation

COFE provides an unusual facility by which the user can condense the display of "paragraphs" in the body of a memo [ref: Arch]. For example, consider the following lines in the body of a memo:

```
         1:          Weekly Schedule 12/6/82
         2:
Mon       : Request status reports from project directors.
         4: Have Williams prepare report instead of Anderson.
         5:      (Return O'Brien's draft to him.)
         6:
         7: Attend budget meeting at 3 in Conf B.
         8:
         9: Pick up tickets to LA.
        10:    Check on room at Tower Club.
Tues      : Review of Kennedy's speech filing system at 10:30.
        12: (Ask about condensation ratios and transmission rates.)
        13: Check on delivery of new filters.
        14:
```

Figure 3. Sample Memo Prior to Condensation

After condensing three of these paragraphs, the screen would appear as follows:

```
        1:          Weekly Schedule 12/6/82
        2:
Mon      : Request status reports from project directors. . . .
        7: Attend budget meeting at 3 in Conf B.  . . .
        9: Pickup tickets to LA.  . . .
Tues     : Review of Kennedy's speech filing system at 10:30.
       12: (Ask about condensation ratios and transmission rates.)
       13: Check on delivery of new filters.
       14:
       15: Leave for airport not later than 3:30.
       16:
Wed      : Out of office.
       18:
Thurs    : Out of office.
```

Figure 4. Sample Memo After Partial Condensation

For the purpose of condensation, a paragraph is defined as follows:

(a) each non-blank line following a blank line begins a paragraph

(b) each line with a user-supplied line-name begins a paragraph

The **condense/expand** command reverses the form of the paragraph denoted by the screen cursor -- condenses a normally displayed paragraph to a single line, or expands a condensed paragraph to normal form. Condensation is relatively permanent -- a condensed paragraph remains condensed over multiple viewings and even multiple sessions, until the user explicitly expands it.

8.1. Annotation of a Memo

COFE also has a special form of condensation that is especially useful for attaching supplemental notes to individual lines of a memo. For example, the following memo has notes attached to lines 4 and 9, the former shown in expanded form, and the latter condensed.

```
NAME: budget meeting                                  LAST USED 11/25/82
  KEPT IN FOLDER: williams    OF FILE: mail           RETAIN TO 12/25/82
MEMO NBR 682       LINES 24   (protected)             ALERT ON
------------------------------------------------------------------------
To      : All Department Heads
Subject : Budget Review
      3:
      4+ There will be a meeting of all departments heads and
NOTE        (Note inserted 12/2/82 09:43 by rwc)
NOTE     John -- I will be away. Can you attend in my place?
NOTE     You will need project summaries and my draft budget.
      5: project managers in my office on Friday 12/10 at
      6: 10 AM.
      7:
      8: We must have final budgets ready for division approval
      9+ not later than 12/17.
     10:
     11: The guidelines for general expense categories are the
------------------------------------------------------------------------
COMMAND:
DESC: of mail
TRAIL: 683 <<682>>
```

Figure 4. Sample Memo with Expanded and Condensed Notes

Each note begins with a header line, automatically supplied by the system. These headers are particularly useful in identifying the origin of notes on memos that have been forwarded by one recipient to another. Each recipient can recognize both the chronology and authorship of text and notes on a memo that is circulated among a number of users.

In condensed form, the existence of a note is indicated only by the "+" character in position 9 of the line. In expanded form, the body of the note and its header line are inserted following the noted line. In expanded form, the note can be edited in the same way as normal text lines.

As noted in Section 6, an important use of annotation is to allow identifiable comments on memos originating as mail.

8.1.1. Non-character Notes

In the existing COFE prototype, notes are entered from a keyboard, just like other memo content. However, in a version under development, notes can also be entered by handwriting on a digitizing tablet. When a handwritten note is **expanded**, the digitized image is displayed in a window of the screen.

There are also other ways in which a digitized-image note could be introduced into the system. A xerographic "page scanner" could introduce an image of arbitrary pages -- including pictures or graphs, as well as printed text. Such an image could be stored, retrieved, forwarded, etc., as a note without requiring any interpretation of the content by the system. Similarly, notes could be introduced in spoken form, and used without addressing the difficult problem of voice recognition. Both of these options are being considered in an advanced version of COFE.

9. Action History

A history of significant actions is automatically maintained for each memo. The first event recorded is the creation of the memo. Each subsequent change, or transmission of the memo is appended. This history is displayed, in place of the memo body, by the **history** command. For example:

```
NAME: program standards                          LAST USED 4/2/82
  KEPT IN FOLDER: comp6    OF FILE               RETAIN TO 12/31/82
MEMO NBR 3079        LINES 37                    ALERT ON
________________history__________________________________________
4/2/82     08:12   Used in creation of memo NBR 4021
4/1/82     15:29   Accepted by JEH
3/29/82    10:05   ReNAMEd and relocated
3/27/82    14:02   Unsent to FBS
3/27/82    11:04   Rejected by DG
3/27/82    10:01   Content modified
3/26/82    13:29   Sent to DG
3/26/82    13:27   Sent to FBS
3/26/82    13:22   Sent to JEH
3/24/82    07:39   Content modified
3/24/82    07:32   Created, copying memo NBR 48

_________________________________________________________________
COMMAND:
DESC: program standards
TRAIL: <<3079>>
```

Figure 6. Sample History of a Memo

10. Protection and Security

COFE is a system of private, single-user libraries and has no generalized facilities for shared access to a single library. The owner of a memo can make it accessible to other users by one of three means:

1. **send** a copy to specific recipients,
2. **send** a copy to the special "notices" user,
3. position the memo in the "public" FILE from which it can be viewed and copied by other users.

COFE also has modest facilities to protect a user against the consequences of his own mistakes:

The hazard of accidental removal is reduced by requiring that the **remove** command be repeated, to confirm the intent to remove a memo.

A memo can be **protect**ed against alteration. While the memo can subsequently be **unprotected**, this requires an deliberate action by the user, so the protected state reduces the chances of accidental modification.

The current memo can be **restore**d to the state that existed when it last became current. This means that a botched-up change can be discarded and started over. However, note that a memo can be **restore**d only while it is still current.

11. The Archival Facility

Eventually, a user's library grows to the point where a request for space (to create a new memo or extend an old one) cannot be satisfied from unused space. When this occurs, the system automatically archives one or more memos to free the necessary space.

The memo selected to be archived is the oldest (by LAST USED date) memo for which the RETAIN date has past. If there is no memo for which the RETAIN date is past, the system appeals to the user to release space by manually **remov**ing some memo(s).

Eligibility for archiving depends on the RETAIN date, which depends on the user's "automatic retention interval" (a value listed in his "controls" memo -- NBR 7). Say the retention interval is r days. The system ensures that the RETAIN date is at least LAST USED + r each time LAST USED is changed, but the user can also manually set the RETAIN date.

Archiving works by sending a copy of the memo to a special user named "archives" and then removing the original from the user's library. Acceptance of the memo at archives is automatic and immediate, and an entry is made automatically in the user's "mail to archives" memo (NBR 8). Of course, the library at archives also has finite capacity, so eventually the memo must be **remove**d from that library as well, at which point the corresponding entry in "mail to archives" is deleted. In effect, archives is simply an interim state on the way to absolute removal. While at archives, a memo can, with some effort, be retrieved and reestablished in the local library. But once the memo has been removed from archives, COFE can no longer retrieve it.

Memos at archives can be viewed by **display** with a description specifying "AT archives" (see Section 4.1.3). An archived memo can effectively be reestablished in the local library by **create** with an argument describing the archived memo (see Section 3.1.1). (This creates a new memo, copying the body of the archived memo.)

12. The Help Facility

The help facility in COFE is simply a collection of builtin memos containing instructions and explanations. The NAME of each of these memos includes the word "help" and they are located in the FOLDER "help" in the FILE "builtin". Consequently, the **help** command is functionally equivalent to

display help

except that the trail for this display task always begins with builtin memo NBR 8. This memo is a "help directory" that lists the NAMEs, NBRs and subjects of other help memos. From this starting point, the user can display other help memos directly by NAME or NBR, or can browse with **next** and **previous.**

Although the builtin help memos are protected against change or removal, the user can arbitrarily augment this facility with additional help memos.

13. Acknowledgements

Many people have contributed to COFE. Daniel DeJohn and John Tais participated in its design and, with Risto Bell and Susan Manning, in the prototype implementation. Some features of COFE can be traced to an earlier system called PIER developed by Conway and Worona in collaboration with Dr. Jack Rosenfeld of IBM Research. We are also indebted to users of PIER who made many helpful suggestions.

14. References

[Arch] Archer, J., Jr. and R. Conway, "Display Condensation of Program Text", IEEE Transaction on Software Engineering, Vol. SE-8, No. 5, (Sept. 82) pp. 526-529

[Meyr] Meyrowitz, N., and A. van Dam, "Interactive Editing Systems", ACM Computing Surveys, Vol. 14, No. 3 (Sept. 82) pp. 321-416

APPENDIX A: Summary of Commands

Commands are described below using the following notation:

boldface words are commands

* indicates a "toggle" command -- repetition constitutes an inverse

[ ] indicates an optional argument

"integer" is a positive integer

"description" is text describing memo(s) (see Appendix B)

"recipient" specifies recipient of mail; either
  a name or nickname in "mail addresses",
  the name of a list memo, or
  a system address

I. Commands can be considered in three groups:

1. "Basic" commands, supported by dedicated keys on most keyboards:
   - **return** (or **enter**)
   - **up, down, left, right**
   - scroll **forward**, scroll **backward**
   - tab
   - erase character, clear to end-of-line

2. "Primary" commands -- the most important and frequently used commands:
   - **accept mail** [integer]
   - **create new memo** [description]
   - **display memo** [description]
   - **help**
   - **insert line(s)** [integer]
   - **list memos on trail** [description]
   - **next memo on trail**
   - **send memo(s)** [recipient]

3. "Secondary" commands (less frequently used):
   - ***condense paragraph**
   - **copy lines** [description] [line range]
   - **delete line(s)** [integer]
   - ***history**
   - ***mark line**
   - ***note condense**
   - **note insert line(s)** [integer]
   - **previous memo on trail**
   - **print memo(s)**
   - ***protect memo**
   - **quit session**
   - **remove memo**
   - **restore memo**
   - **resume prior task**
   - **use memo**

II. The commands are grouped functionally, and described in more detail below:

Creation and removal of memos:

create [description]
create a new memo; make it current; if description is given copy the newest memo satisfying description into the newly created memo (see Section 3.1.1)

remove
remove the current memo; (confirmation is requested to avoid accidents)
--builtin memos cannot be **remove**d
--except for mail memos, which can be **remove**d directly, **protect**ed memos must be **unprotect**ed before they can be **remove**d

Retrieval of memos:

display integer
make the memo whose NBR is specified the current memo

display [description]
interrupt the current display task and begin a new one; make the newest memo satisfying the description the current memo (see Sections 4 and 7); if no description is given, all memos satisfy

next
make the next older memo satisfying the description of the current display task the current memo

previous
move back to the last previous memo retrieved by the current display task

resume
resume execution of the most recently interrupted display task (see Section 7)

list [description]
create a new "list memo" containing a list of memos satisfying the description; if description is omitted, use the description of the current display task, continuing from the current memo (see Section 4.2)

use
update the LAST USED field of the current memo to today's date (done automatically when memo is modified, mailed, or printed)

***history/text**
display the (most recent) history of the current memo (instead of the usual text window) (see Section 9)

help
equivalent to "**display** help" (see Section 12)

Communication:

send recipient
send a copy of the current memo to the indicated recipient, except that if the current memo is a "list" all the memos described in that list are sent; log

each memo sent in "outgoing mail" of the sender and "incoming mail" of the recipient (see Section 6)

send description

like single recipient **send**, except that the newest memo satisfying the description is interpreted as a list of recipients

accept [position]

create a new memo equivalent to the memo in the indicated position in "incoming mail"; if position is not specified, accept the most recently arrived mail; delete corresponding entries from "incoming mail" and the sender's "outgoing mail"; append entry to sender's "accepted mail" (see Section 6)

delete [position]

deletion of entry in "incoming mail" rejects the associated mail

deletion of entry in "outgoing mail" recovers mail that has been sent but not yet accepted

(see Section 6)

print

print a copy of the current memo, except that if the current memo is a "list" all the memos described in that list are printed

Condensation:

***condense/expand**

condense the display of the indicated paragraph from normal form (full lines) to single-line form; expand the condensed paragraph to normal form (see Section 8)

***note condense/note expand**

condense the indicated note so that only a "+" on the noted line appears; expand a condensed note to full display form (see Section 8.1)

Annotation:

note insert [integer]

insert "integer" blank lines in a note immediately after the cursor line, creating a new note (with a header) if none exists at this point, or expanding an existing note (see Section 8.1); default integer is 1

Editing and cursor-positioning commands:

Characters are entered at the cursor position on the screen; the cursor is automatically advanced.

erase character

erase the single character at the cursor; all characters to the right of the cursor are moved one position to the left

clear to end

clear all characters from the cursor position to the right end of the field

tab

move the cursor forward to the next tab stop

return
move the cursor forward to the next unprotected field; or, if the COMMAND field contains a command, execute it

window
move the cursor forward to the next window (the cycle is: header, text, command windows)

up, down
move the cursor up/down 1 line;
cursor can be moved **up** into the text window from the command window or **down** into the text window from the header window, but not out of the text window with **up/down**; **up/down** at limit of text window causes scrolling of the text window

left, right
move the cursor left/right 1 position

forward, backward [integer]
scroll the text window "integer" lines toward the end/beginning of the memo; default integer is half the length of the text window

insert [integer]
insert "integer" blank lines immediately after the cursor line, renumbering following lines; default integer is 1; position the cursor on first inserted line

delete [integer]
delete "integer" lines beginning with the cursor line, renumbering following lines; default integer is 1 (**delete** also has special use in mail facility)

***mark/unmark**
mark the cursor line (if unmarked) as limit of copy operation; copy ranges from first marked line to second marked line (inclusive); **mark** reverses the mark-state of a line: an unmarked line becomes marked, a marked line becomes unmarked; mark is shown between line name and line text (column 10 in most implementations); only one memo can be marked at any instant; at most two lines can be marked in a memo

copy [description] [line range]
copy segment of newest memo satisfying "description" in an insertion before the cursor line in current memo; if description is omitted source is also current memo; **mark**ed lines may be used instead of "line range"; if line range is omitted and no lines are marked, entire memo is copied

restore
restore the current object to the state that existed when it most recently became current

Termination:

quit
terminate the session; the next session begins in precisely the same state the previous session terminated

APPENDIX A.1: Command Entry

Command Entry Using Special Function Keys:

When special function keys are available, commands are assigned to particular keys. The key assignment is shown on the bottom line of the screen. Since the number of (primary and secondary) commands exceeds the number of special keys available, several commands are assigned to each key.

A shifting scheme is used to distinguish between the different commands assigned to each key. That is, each key has a normal (unshifted) value, corresponding to a primary command. It also has shifted values, corresponding to the secondary commands. As far as possible, the different commands assigned to a single key are logically related to each other.

> Two special keys serve as command shift keys. A command shift key alters the key assignments, and causes the new assignments to be displayed on the screen. The duration of the shift is a single command. That is, assignments automatically revert to the normal (primary) value after a command is given.

When 10 special keys are available, the following assignments are used:

key	Normal (primary)	Shift 1 (secondary)	Shift 2
1	shift 1	normal	shift 1
2	shift 2	shift 2	normal
3	**insert**	**delete**	**note insert**
4	**display**	**condense**	**note condense**
5	**next**	**previous**	**history**
6	**list**	**resume**	**use**
7	**send**	**print**	
8	**accept**	**mark**	**copy**
9	**create**	**remove**	**protect**
10	**help**	**restore**	**quit**

The current command key assignment is shown on the KEYS line at the bottom of the screen. The nominal prompt lines are the following:

Normal (unshifted):

1shift 2shift 3inse 4disp 5next 6list 7send 8acce 9crea 10help

Shift level 1:

1norm 2shift 3dele 4cond 5prev 6resu 7prin 8mark 9remo 10rest

Shift level 2:

1shift 2norm 3nins 4ncon 5hist 6use 7 8copy 9prot 10quit

The actual prompt line also indicates which commands are usable in the current state by emphasizing those command keys with a distinctive font (brightening, backlighting, etc.) For example, suppose the current state is the following:

(a) the current memo is protected
(b) there is no older memo satisfying the current description
(c) the "incoming mail" memo is empty.

This means that neither **insert**, **next** nor **accept** is currently usable, hence the normal (unshifted) prompt line would appear as follows:

1shift **2shift** 3inse **4disp** 5next **6list** **7send** 8acce **9crea** **10help**

The following commands are always usable, hence are always emphasized when they appear on the prompt line:

1shift
2shift
display
list
send
create
help

All other commands have various contextual restrictions on their use, listed below:

insert -- there must be a current memo, unprotected, with cursor in text window
next -- there must be an older memo that satisfies the current description (the system looks ahead)
send -- there must be a current memo
accept -- the "incoming mail" memo must not be empty
delete -- there must be a current memo, unprotected, with cursor in text window
condense -- there must be a current memo, with cursor in text window
previous -- there must be a prior memo on the current trail
resume -- there must be an interrupted display task
mark -- there must be a current memo, with cursor in text window, containing at most a single mark, with no other memo containing any mark
remove -- there must be a current memo, unprotected
restore -- some change must have been made in the current memo since it became current

note insert -- there must be a current memo, with cursor in text window

note condense -- there must be a current memo, with cursor on a line in a note or a line with a note attached

history -- current memo must be a memo with history (that is, not "builtin")

use -- the LAST USED date of current memo must be different from current date

print -- there must be a current memo

copy -- there must be a current memo, unprotected, with cursor in text window

protect -- there must be a current memo, for which the protect status can be changed

If the user gives a command that is not currently usable (not emphasized in the prompt line), the response is an explanatory message. For example, if **accept** is given when "incoming mail" is empty, the response is the message:

"Sorry, there is no incoming mail to ACCEPT."

Textual Command Entry:

Commands may also be entered textually in the command window. For example,

CMD: display smith letter **return**

is equivalent to:

CMD: smith letter **display**

When special function keys are available, the user can employ either key or keyword entry. Obviously, on keyboards without special function keys textual entry must be used.

When textual entry of commands is employed, it is not necessary to enter the entire keyword. Any unambiguous prefix of the keyword is sufficient.

APPENDIX B: Memo Descriptions

```
<description> ::= <numeric description> |
                  <attribute description>

<numeric description> ::= <memo NBR>

<attribute description> ::= [<NAME spec>]
                            [<FOLDER spec>]
                            [<FILE spec>]
                            [<remote library spec>]
                            [<date spec>]
                            [<content spec>]

<NAME spec> ::= <string list>

<string list> ::= <string> | <string> <string list>

<string> ::= <word> | <quoted string>

<FOLDER spec> ::= IN <word>

<FILE spec> ::= OF <word>

<remote library spec> ::= AT <user name>

<date spec> ::= [RETAIN|ALERT] ON <date> |
                [RETAIN|ALERT] BEFORE <date> |
                [RETAIN|ALERT] AFTER <date>

<date> ::= mm/dd/yy | mm/dd

<content spec> ::= WITH <string list>
```

APPENDIX C: Summary of Builtin Memos, Special Users and Locations

Builtin Memos

```
NAME tasks
    KEPT IN FOLDER lists OF FILE builtin
    MEMO NBR 1  (protected)
        Stack of interrupted display tasks; used by resume.

NAME outgoing mail
    KEPT IN FOLDER lists OF FILE builtin
    MEMO NBR 2  (protected -- but user can delete lines)
        List of mail sent but not yet accepted by recipient.

NAME incoming mail
    KEPT IN FOLDER lists OF FILE builtin
    MEMO NBR 3  (protected -- but user can delete lines)
        List of mail received but not yet accepted
        or rejected.

NAME accepted mail
    KEPT IN FOLDER lists OF FILE builtin
    MEMO NBR 4  (protected -- but user can delete lines)
        List of mail accepted by recipient.

NAME rejected mail
    KEPT IN FOLDER lists OF FILE builtin
    MEMO NBR 5  (protected -- but user can delete lines)
        List of memos deleted from recipient's incoming mail.

NAME mail addresses
    KEPT IN FOLDER lists OF FILE builtin
    MEMO NBR 6
        List of names and system addresses for mail delivery.

NAME mail to archives
    KEPT IN FOLDER lists OF FILE builtin
    MEMO NBR 7  (protected -- but user can delete lines)
        List of memos sent to archives.

NAME controls
    KEPT IN FOLDER lists OF FILE builtin
    MEMO NBR 8 (partially protected -- user can change lines, but
                not line-names)
        List of options and control values:
            1. Automatic retention interval - used to set RETAIN
               date.
            2.

NAME help directory
    KEPT IN FOLDER help  OF FILE builtin
```

MEMO NBR 9
Table of contents of "help" memos with initial instructions.

NAME help <subject>
KEPT IN FOLDER help OF FILE builtin
MEMO NBR 10 - ?
Instructions and explanations of various facilities.

Special Users

Notices
This user is the recipient of public notices. All FILEs at notices are public. Acceptance of all mail by notices is automatic and immediate.

Archives
This user is the recipient of archived memos. Acceptance of all mail by archives is automatic and immediate, and it is automatically acknowledged by an entry in the sender's "mail to archives" memo. Each memo at archives is identified with the contributing user, and is accessible only to that user.

Special Folders and Files

FOLDER <sender> OF FILE mail
Incoming mail, when **accept**ed, is placed in the FOLDER with the name of the sender, of the "mail" FILE. Since mail is protected against change, it necessarily remains in this FOLDER and FILE until **remove**d or archived.

FILE public
The "public" FILE of any library is accessible to other users. Public memos may be viewed (by **display**) or copied (by **create**), but cannot be altered by other users.

FOLDER lists
When a memo in the "lists" FOLDER are sent or printed, it is treated as a list of memos: each memo on the list is sent or printed, rather than the list itself. Memos created by the **list** command are automatically in the "lists" FOLDER.