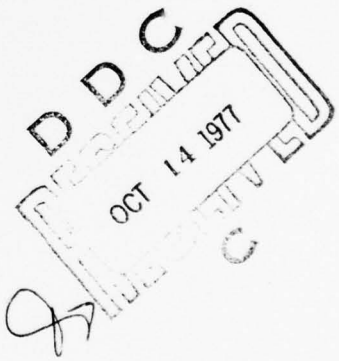
Bolt Beranek and Newman Inc.

12

Report No. 3618

Message Technology Research and Development

Quarterly Progress Report No. 6
2 April 1977 to 2 July 1977



September 1977

Submitted to:
Defense Advanced Research Projects Agency

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: BBN REPORT NO. 3618
2. GOVT ACCESSION NO.
3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle): MESSAGE TECHNOLOGY RESEARCH AND DEVELOPMENT.
5. TYPE OF REPORT & PERIOD COVERED: 4/2/77 - 7/2/77
6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s): J. Burchfiel, T. Myer
8. CONTRACT OR GRANT NUMBER(s): MDA903-76-C-0212, ARPA Order-3161
9. PERFORMING ORGANIZATION NAME AND ADDRESS: Bolt Beranek and Newman Inc. 50 Moulton Street Cambridge, Massachusetts 02138
10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS
12. REPORT DATE: September 1977
13. NUMBER OF PAGES: 27 + iii
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office): Quarterly progress rpt. no. 6, 2 Apr - 2 Jul 77
15. SECURITY CLASS. (of this report): Unclassified
15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Distribution of this document is unlimited. It may be released to the Clearinghouse, Department of Commerce for sale to the general public.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

This research was supported by the Defense Advanced Research Projects Agency under ARPA Order No. 3161

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

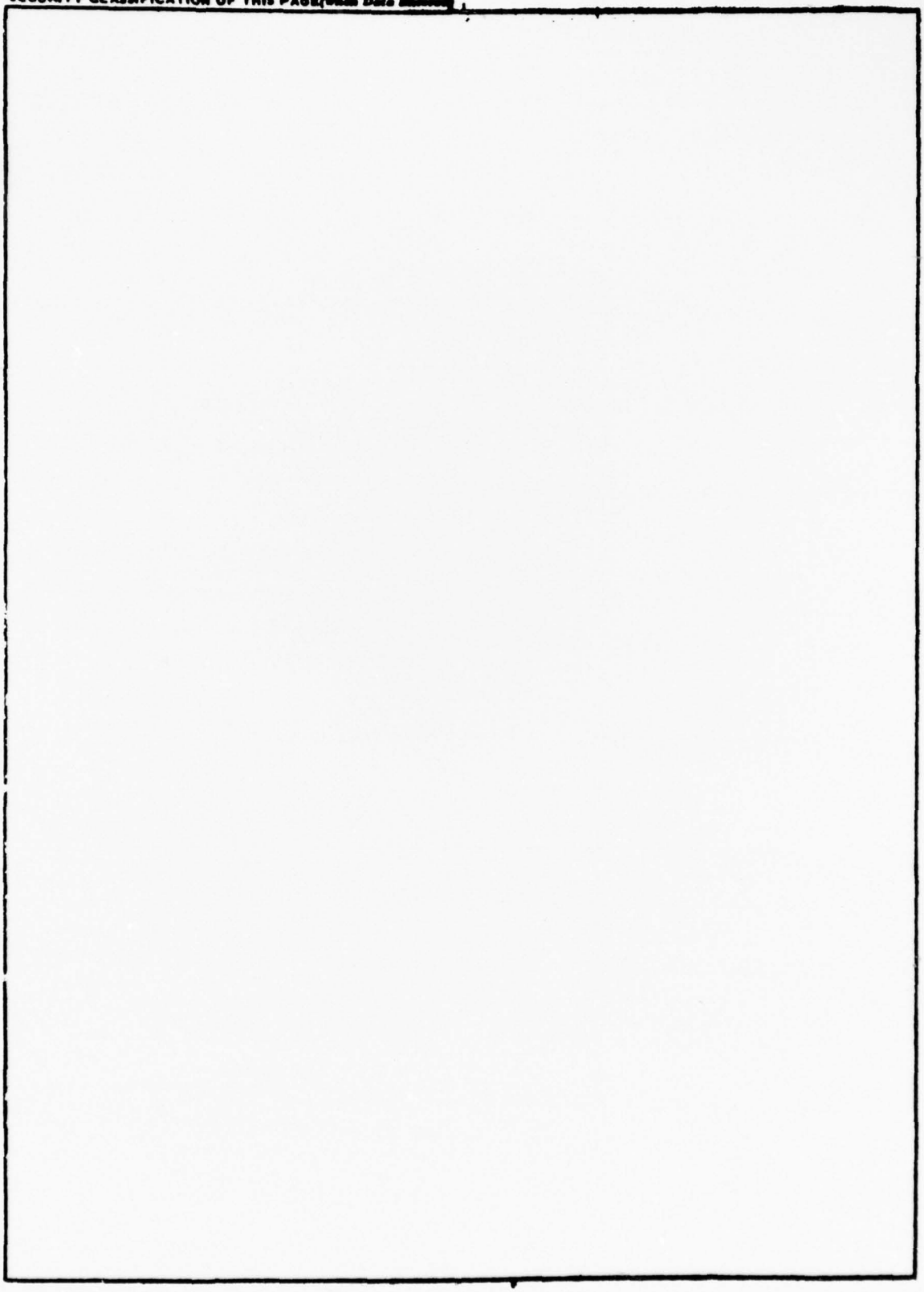
Hermes
Message Processing
Tenex Security
CINCPAC Test

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This report describes BBN efforts in the continuing development of the HERMES message-processing system, with respect to system design, security requirements and preparations for the DARPA/NAVY/CINCPAC interactive test.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

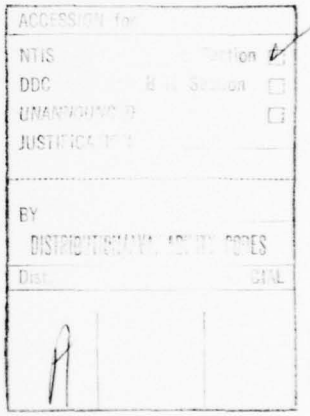


BBN REPORT NO. 3618 September 1977

MESSAGE TECHNOLOGY RESEARCH AND DEVELOPMENT

Quarterly Progress Report No. 6

2 April 1977 to 2 July 1977



The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied of the Defense Advanced Research Projects Agency of the United States Government.

This research was supported by the Defense Advanced Research Projects Agency under ARPA Order No. 3161 Contract No. MDA903-76-C-0212

Distribution of this document is unlimited. It may be released to the Clearinghouse, Department of Commerce for sale to the general public.

TABLE OF CONTENTS

1. INTRODUCTION

This report covers progress in message technology under the contract "Message Technology Research and Development" for the period 2 April through 2 July 1977.

This work is a continuation of work on message technology performed under the ARPA Contract MDA903-76-C-0212 "Distributed Computation and TENEX Related Activities" during 1975.

During the April through June quarter, following the termination of our work on the version of HERMES which we demonstrated for the DARPA/NAVY/CINCPAC test (the Military Message Experiment or MME), we implemented and released "civilian" version of HERMES incorporating many of the features developed for the MME but not including MME security features.

HERMES 4.0

We installed a new "generation" of HERMES when we implemented and released HERMES 4.0 for general use by DARCOM and other users through the ARPANET on the BBN host computers. HERMES 4.0 was also installed on the SRI-KA and SRI-KL host computers, under control of an access list, and on the ISIA computer. (SRI-KL replaced the OFFICE-1 host computer on the ARPANET during this quarter.)

Successive versions of HERMES 4.0 provided a number of new features and improvements during this quarter, in addition to correcting bugs as they were uncovered in HERMES operations.

Features Incorporated from MME-HERMES

Several features first implemented in MME-HERMES were installed in HERMES 4.0. These include the ability to process extremely large message-files, the provision (under switch control) of file-handling capabilities entirely with HERMES, the REDISTRIBUTE and REFILE commands, the TERMINAL-LOCK feature and the availability of the WE Scope Editor for HP2640/45 terminals.

Improvements in Speed

This quarter witnessed a dramatic improvement in the speed of HEERMES operations. This was accomplished in part through the incorporation in HERMES 4.0 of the improved implementation of the auxiliary (ParSeq) files used to aid the handling and searching of message-files. This implementation was refined and improved in successive versions of HERMES 4.0.

Another factor was a detailed review of HERMES code to find and correct the places where the speed of the operating code could be improved.

Tests conducted in May 1977 showed HERMES 4.0.12 to be faster than HERMES 3.5 but slower than two comparison systems, MSG and

HG. Preliminary tests at the end of June indicate that HERMES 4.0.17 now closely approaches MSG in speed.

Fast Local Sending

In response to requests from users, we have modified HERMES to give the user the option of appending messages directly to MESSAGE.TXT or MAIL.TXT files where directory protection allows the user to write into the addresses's directory.

We expect this provision to be temporary. The MAILER program ought to be rewritten to improve its performance for both local and remote sending so that there will no longer be a significant difference in the length of time it takes for immediate local sending through MAILER compared with local sending by direct appending.

Other New Features

We have improved the performance of "Ask" setting of the UPDATE-PROFILE switch, which controls the automatic profile feature.

The user now has the option of being included or omitted in the CC field when the REPLY command generates an automatic set of header-fields.

Two new fixed templates, a new version of PRINT-FORM which suppressed redundant header information and a LONG-PRINT-FORM,

which prints the message exactly as received, have been added. SHOW, NEWS and HELP can now be output to the lineprinter or to a file. This means that all commands capable of printing output on the user's terminal can be redirected to another "destination" such as a file or the line printer. Information stored in the auxiliary (ParSeq) files is now protected in the event that other portions of the auxiliary file must be automatically discarded because of failure in the operating system or other errors.

Other minor changes have been made to make HERMES work more smoothly.

HERMES in Data Management

As users have become familiar with the user-field and template-creation facilities of HERMES. Some groups of BBN personnel outside the HERMES group, and working on unrelated projects, have chosen to use HERMES to solve their data management problems. One group has created a file of about 100 messages containing decriptions of programs and computerized courses relating to computer aided instruction for the handicapped. Another group has in the planning stage, the computerization of a large file of literature references related to underwater technology. A third group has demonstrated how HERMES can be combined with the BBN report preparation program MRUNOFF to facilitate the preparation of business-type form letters and envelopes from mailing lists maintained as HERMES message-files.

In addition, Project HERMES itself maintains a growing file, which currently contains approximately 100 descriptions of literature references related to message technology.

Statistics on HERMES Use

We have continued to collect and analyze statistics on HERMES usage. As of the end of June, HERMES was being used in approximately 2000 individual sesions per week.

The LDMX-TENEX Interface

The translator between LDMX and TENEX messages was completed and installed for the MME, and was accepted by the Navy.

Future Plans

We have divided our efforts and future message systems into three parallel projects:

1. To implement and release HERMES 4.1 as a reliable production system with a self-consistent set of improved functions but without major novel features.

2. To study and begin implementation of a system operating in an environment of distributed computer architechture, involving minicomputers, microprocessors, and "intelligent" terminals, as well as large-scale computers, including requirements for military security. (Time scale 1 to 2 years)

3. To design and develop a "structured" laboratory to study the functional and human interface aspects of message systems and to further study the potential of distributed architecture, with emphasis on advanced terminals and graphic display.

2. HERMES 4.0

We installed a new "generation" of HERMES when we implemented and released HERMES 4.0 for general use by DARCOM and other users through the ARPANET on the BBN host computers. HERMES 4.0 was also installed on the SRI-KA and SRI-KL host computers, under control of an access list, and on the ISIA host computer.

2.1 HERMES 4.0.1

HERMES 4.0.1, released on 2 May 1977 differed from the previous HERMES 3.5 in a number of major features and achieved a considerable improvement in speed of operation.

2.1.1 Ability to Process Very Large Message-Files

The new implementation of the auxiliary "parseq" files used to aid the handling and searching of message-files means that, in practice, the size of the message-file is now limited by the disk space available. The theoretical upper limit of message-file size is one million messages. This contrasts with the former limit of about two thousand messages.

2.1.2 HERMES File Names

Under control of the FILENAME-INPUT switch, HERMES files can be handled like other objects. They can be addressed with single-word names that take advantage of the ?, <ESC> and spelling correction; they can also be shown or deleted within HERMES. The auxilliary PARSEQ (up-arrow) files are dealt with automatically by HERMES.

FILENAME-INPUT TENEX [X]HERMES

1. TENEX: File-names follow TENEX or TOPS-20 conventions, and must be deleted outside HERMES, through the use of the operating system.

2. HERMES: file-handling can be accomplished entirely within HERMES. File-names act like HERMES objects, have the form <DIRECTORY>NAME. In the connected directory, <DIRECTORY> may be omitted. Files are deleted with the MESSAGE-FILE-EDITOR and the DRAFT-FILE-EDITOR. The names TERMINAL and LINE-PRINTER can be used as synonyms for TTY: and LPT:.

2.1.3 The REDISTRIBUTE Command

Syntax: >REDISTRIBUTE <message-no> <CR>
Defaults: CMESSAGE

HERMES allows you to redistribute a message without wrapping it up in a jacket (like FORWARD), or changing the Sender:, Date:, and Message-ID fields (like EXPLODE).

REDISTRIBUTE prompts you for addresses that the message is to be redistributed to, and then asks whether to refile the message. Three new fields are added:

Redistributed-To: <addressee-list>
Redistributed-By: <connected directory> <login directory>
Redistributed-Date: <date>

2.1.4 The REFILE Command

Syntax: >>REFILE <message-no./end><CR>
Defaults: <message previously exploded>

REFILE places the contents of the CDRAFT (in the message editor) in the <message-no.> slot in the current message-file. You are shown a survey of the message previously in the <message-no.> slot, and asked for your OK before it is deleted.

2.1.5 NEWS, HELP and SHOW Take Destinations

NEWS and HELP now each take a destination which may be, TTY: (or TERMINAL), or LPT: (or LINE-PRINTER), or a file-name.. You can print them on your terminal, on the line-printer or into a file.

Syntax: >NEWS <destination><CR>
Default: Terminal or TTY:

```
Syntax:   >HELP <destination><CR>
Default:          Terminal or TTY:
```

The SHOW Command is similar, at top command level:

```
Syntax:   >SHOW <object/object type> <destination><CR>
Defaults:               All              TERMINAL or TTY:
or
Syntax:   >SHOW <names/All> <object-type> <destination><CR>
Defaults:             All      Objects     TERMINAL or TTY:
```

2.1.6 The TERMINAL-LOCK Feature

<CTRL-Y> calls a program that allows you to lock your terminal without logging out.

```
Syntax:  ><CTRL-Y>
         KEY TO UNLOCK CONSOLE IS: <key><CR>
         PASSWORD: <user's password><CR>
```

Your terminal is now locked. To unlock the terminal, type the key.

You can abort the program after <CTRL-Y> and before <CR>:

On TENEX, type <RUBOUT> or <CTRL-Q>
On TOPS-20, type <CTRL-U> or <CTRL-Q>.

If the key does not work the first time you try it, there are probably extra characters in the buffer, typed while the terminal was locked. Type <RUBOUT> (on TENEX) or <CTRL-U> (on TOPS-20) and try again.

If you forget the key, you must go to another terminal, log in under another job number, and then log out the job attached to the locked terminal.

2.1.7 The Automatic Profile Feature.

The "Ask" setting on the UPDATE-PROFILE switch now asks only when you have actually made changes in your filters, templates, switch settings or user-fields.

2.2 HERMES 4.0.5, 4.0.8 and 4.0.12

HERMES 4.0.5, released on 3 May 1977 introduced two new features.

2.2.1 The REPLY-INCLUDEME Switch

The new REPLY-INCLUDEME switch controls whether your name is included in the CC:-field of the Reply message if it appears in the To:- or CC:-field of the message being replied to. The switch is initially set to "Ask". If you wish to continue to have your name included, whenever it appears in the original message, as in earlier versions of HERMES, set the switch to "Yes". If you want your name excluded, set to "No".

HERMES asks whether to include your name only if the name actually appears in the message being replied to, and shows you the exact form of the name used.

```
>REPLY 23<CR>
Include SMITH in the CC field?
```

2.2.2 Sequences Safeguarded Under Error Conditions

We have improved the way HERMES handles error conditions in the "parseq" (Up-arrow) files which act as companions to the message files, and which hold information about the structure of the files (the PARse), as well as storing named SEQuences. If the parse information is wrong, HERMES discards it, and generates a new parse, preserving the sequences intact.

HERMES 4.0.8 and HERMES 4.0.12 also released in May, included a number of bug fixes.

2.3 HERMES 4.0.17

The next release, 4.0.17, on June 25, added two major new features, in addition to bug fixes:

2.3.1 Fast Local Sending

The TRANSMIT-NOW switch now has four positions: "Ask", "Yes(All)", "LocalOnly" and "No(Queue)".

"LocalOnly" transmits local messages by appending directly to the MESSAGE.TXT or MAIL.TXT file, where directory protection allows the user to write into the addressee's directory. Other messages are queued for MAILER.

Note: The "Ask" position still queries for invoking MAILER immediately. However, you can type subcommands that give the results of any switch setters.

We expect that this provision will be temporary; MAILER ought to be rewritten to improve its performance for both local and remote sending so that local sending through HERMES will not be necessary.

2.3.2 New Printing Templates

Two new templates, PRINT-FORM and LONG-PRINT-FORM, have been added to the collection of fixed templates, replacing the old PRINT-FORM.

```
PRINT-FORM
(1)  Message-No.:+ Char-Count: Rcvd-Date: "  " Status:+
(2)  From:+
(3)  To:+
(4)  CC:+
(5)  Bcc:+
(6)  Subject:+
(7)  In-Reply-To:+
(8)
(9)  Text:+
```

```
LONG-PRINT-FORM
(1)  Message-No.:+ "; " Char-Count:+ Status:+
(2)  Verbatim:
```

The "Verbatim:" item in LONG-PRINT-FORM prints the message exactly as received, at a 5-to-6 percent saving in printing time over the "Standard:", item used in the old PRINT-FORM template. "Standard:", which is still available, is a template item that prints all the fields in the order of the STANDARD template. (The "Other:" item has been moved to the top of "Standard:" so that the "Mail from ..." line will appear at the top of the message.)

Minor improvements in HERMES 4.0.17 have been implemented to make the program work more smoothly for the user.

2.3.3 Control-C in a Lower Fork

<CTRL-C> can now be used to escape from certain programs running in a lower fork under HERMES. If you enter Teco, Xed, Neted, Spell or We from HERMES, or if you use the HERMES Run <file> command to run a program in a lower fork, you can now type <CTRL-C> and return to HERMES, without losing your HERMES status. You will, of course, lose anythng you created in the lower fork.

If you type two <CTRL-C>'s, you will escape from HERMES only, and land in the Exec. You can return to HERMES by typing CONTINUE to the Exec prompt, @.

2.3.4 Control-E Preserves Edits

<CTRL-E> at any prompt level now returns you to the same prompt level and does not abort the edit you are performing. (This has been true of the MESSAGE-EDITOR; it now works consistently throughout HERMES.)

2.3.5 Improved Primary Access to Message-Files

If more than one user accesses a message-file, and the user with primary access leaves, the lock on the message-file is cleared. The next user to access the file becomes the primary user.

2.4 IMPROVEMENTS IN SPEED

We performed the three-way, side-by-side benchmark tests of HERMES (Version 4.0.12), MSG, and HG in May 1977. The results are reported in Appendix A.

HERMES 4.0.12, which is the version that was used in this test has now been superseded by HERMES 4.0.17, which has been considerably speeded up in crucial areas, such as start-up time and message printing. We are working to improve HERMES response time. Preliminary results indicate that it should be possible to increase the speed of HERMES until it is equal to that of MSG. We will report test results in the next progress report.

3. THE USE OF HERMES IN DATA MANAGEMENT

The introduction of user-created message fields makes possible a wide variety of applications of HERMES to data management problems. The first such application was "The Blue File", a database of information on U.S. Navy ships described in the previous progress report. Since this pioneering demonstration, a number of other applications of HERMES have been implemented.

HERMES Literature

Project HERMES has accumulated a group of about 100 literature references related to message technology and has organized them in as a file of HERMES messages.

The template used to COMPOSE the message containing a reference contains an AUTHOR-KEY: field which allows references to be sorted alphabetically by author, while the PUBL-(DATE): field allows sorting by date.

Here is a sample literature reference printed through two different printing templates, SBIBLIO and REF.

```
>print 51 s-biblio<CR>
The Role of Informal Communications in Computer Networks
Message Technology
    51   570 Uhlig, R.P., Martin, S.M., and von Gehren, E.S.
----------

>print 51 ref-biblio<CR>
Uhlig, R.P., Martin, S.M., and von Gehren, E.S.
The Role of Informal Communications in Computer Networks
Pacific Area Computer Network Symposium
Sendai, Japan 21 Aug 1975
```

The applications described next have been produced by BBN personnel outside the HERMES group. They are included as examples of the ways in which HERMES users are exploiting the HERMES data management capabilities.

Computer Aids to the Handicapped

A project on computer aids to the handicapped has created HERMES message-files containing descriptions of system programs and computerized courses. Most of the approximately 100 mesages contain user fields appropriate to the program type, such as users:, curricular-content:, source:, and language-level. Another file, used to keep track of information about computer terminals, contains only five messages, but these are frequently edited and refiled. In all these files, there may also be informal messages that are awaiting editing or reformatting for the purpose of creating messages in the primary format.

Literature on Underwater Technology

The Underwater Technology division of BBN has made a pilot study of the use of HERMES to organize a large file of references to technical literature in that field. They anticipate that the references will be searched primarily on the basis of the information content of the subject matter.

Form Letters from a Combination of HERMES and MRUNOFF

A BBN group that works with the BBN implementation of the MRUNOFF language has developed a method of producing high-quality typewritten letters and other documents. They have implemented a set of printing templates that make use of user-specified fields (of the TEXT and LINE types) within lines containing text-formatting commands in the MRUNOFF language (BBN implementation). When messages are thus printed onto files and then fed as input to the MRUNOFF translator, the result is a formatted file ready for output on a high-quality prrinting terminal. Various letter formats are available, and it is possible to generate form letters automatically from mailing lists that are stored as HERMES message-files.

The feature that distinguishes these projects from ordinary message management is that the messages are usually not sent between different individuals. Instead, units of information are packaged as messages and filed through the use of an FCC field or

the Refile Command. Of central importance is the ability of the HERMES user to create tailor-made message-fields and then to utilize the HERMES template editor to create both composing and printing templates that make use of these individualized message-fields.

Further development might be the expansion of the types of user-fields in HERMES to include numeric fields and the introduction of a facility for producing totals, averages and other arithmetic functions for bookkeeping and budgetary databases.

4. THE IMPLEMENTATION OF HERMES ON THE TOPS-20 MONITOR

Beginning with HERMES 3.5, as reported in the previous Progress report, all versions of HERMES hve been fully convertible between TENEX and TOPS-20 operating systems. The only differences between TENEX and TOPS-20 from the point of view of the user are the following:

	TENEX	TOPS-20
User's INBOX	MESSAGE.TXT;1	MAIL.TXT.1
Other file names	NAME.EXT;VERSION	NAME.EXT.VERSION
Single-character delete	<CTRL-A>	RUBOUT/DELETE key
Command abort	RUBOUT/DELETE key	<CTRL-U>

These changes have been instituted to conform to the conventions adopted by Digital Equipment Corporation, which maintains the TOPS-20 operating system. They have the advantage of conforming to the conventions of DEC operating systems for the PDP-11 and other computers.

As each new version of the HERMES is produced, TENEX and TOPS-20 executive programs are implemented from the same source files.

5. MESSAGE-SYSTEM DESIGN EFFORTS

The design and implementation of future message systems has been split up into three general areas:

1) The completion and refinement of the current HERMES system as a reliable production system on TENEX and TOPS-20 operating systems, for use on the ARPANET.

2) The study of requirements for message-systems one or two years from now, with emphasis on distributed architecture and security requirements.

3) Research into the functionality and human factors aspects of message systems. This includes the development of a breadboard laboratory facility to make it easy to test alternative designs and configurations rapidly.

5.1 The Current Developmental Version of HERMES

We are working on the problem of speeding up the basic operations of the current HERMES. Concurrently, we plan to fill in the gaps in the current command structure, including the provision of templates for the REPLY and FORWARD commands, the enhancement of the current template formatting facility, improved interfacing with other message systems, and improved automatic fix-up of the error conditions and damaged message-files.

5.2 The Architecture of Message Systems

The first computer message systems were of limited functionality. A incoming message could be read and deleted; an outgoing message was composed and sent. Such simplicity did not require a large

amount of computation, nor did it lend itself to much tailoring to the needs of individual users. Thus, the response even on a well-loaded system was fast and the size of the actual software was minimal.

As message systems have grown in functionality they have also grown in need for resources. More computation is done in order to perform more sophisticated actions, and the size of the underlying code has also increased. The performance of today's message systems has suffered because of this. Frequently neither the share of the computer nor the amount of memory available on time-shared computers is sufficient. One of the problems which confront the designers of the next generation of message management systems is the dilemma of response: how can a system be built which has extensive functionality and fast response time without demanding the dedicated use of a large machine?

The answer may lie with the use of the minicomputer and distributed computation technology. Minicomputers are becoming increasingly inexpensive and more powerful. Distribution of the computational tasks should reduce the demands on the central time-shared computer(s) and allow greater flexibility in the front end. We view any system as capable of being split into several types of modules. One or more of these may be combined; one or more may not be needed in a particular system.

5.2.1 The Front End

The Front End of the system, determines the form in which the commands and other system functionalities are presented to the user. Two modules with exclusively front-end functions are the terminal and the display handler.

The terminal: Accepts user input, transfers it to the display handler, accepts and displays output from the display handler. Different terminal types will output different types of information and require different information to control the display. For example, a bit-mapped graphics terminal could make use of windows and sophisticated on-screen editing of text while a standard hard-copy device must deal with information a line at a time.

The display handler: All terminal specific code resides in the display handler. The display handler manipulates special display functions, translates terminal-specific input into terminal independent code and performs command editing functions.

5.2.2 The Intermediate Modules

The modules which process input generated by the Front End are the traffic cop, the command scanner, the grammar and the command expeditor.

The traffic cop: The traffic cop handles I/O to/from the display handler, command scanner, and command expeditor and performs general bookkeeping functions.

The command scanner: The command scanner parses terminal-independent user input according to the rules of the grammar and returns help in the form of defaults, noise words, command completion and spelling correction.

The grammar: The grammar is a set of tables which completely define the syntax of the command language way in which the command gets translated into requests to the backend functions.

The command expeditor: The command expeditor accepts parsed commands from the traffic cop and sees to it that they are properly executed by the backend processes. It handles error codes and handshaking. It forwards output from the backend processes to the traffic cop.

5.2.3 The Backend Processes

The Backend Processes comprise the final module (or group of modules) which does the actual work of the system.

5.3 Research on Functionality and the Human Interface

During this quarter, considerable thought has been given to whether the HERMES message system is an adequate base for carrying out research in message system functionality and human interfaces.

We have, over the past year, been noticing that the ease with which new ideas can be tried out is dropping steadily. The primary reason for this is that HERMES is a production system. While this does ensure that new ideas which get inserted into HERMES will received thorough testing in actual message-processing situations, it has the disadvantage that ideas which will strongly impact the community of HERMES users be thoroughly thought out in advance, that they be carefully integrated into conceptual patterns of existing HERMES, that they be skillfully presented to the user community to prevent misunderstanding. Futhermore, to prevent divergent development, the HERMES systems which are produced and made available to the user community are constrained to be upward-compatible with earlier members of that sequence. Therefore it is not really possible to compare conflicting approaches to meeting user needs. This resulted in making decisions based not on actual experience

with using different systems, but on our best guesses as to how the different systems would "feel"; that is, we were being forced to considerable amounts of "gedanken" research.

The conclusion is that the production environment which surrounds the creation, distribution, and usage of HERMES systems is not a good environment for supporting quick turn-around research into message systems.

We therefore concluded that a different environment was necessary if we were to be able to continue with active research in message systems. We decided to construct a "laboratory" environment to support our research intentions. This environment would have to richly support fast programming efforts, and would not need to produce production-quality code (could be space and time inefficient).

5.3.1 The Experimental laboratory Using INTERLISP

We have begun construction of a research environment based on the INTERLISP programming environment. This richly supportive system is available on TOPS-20, and has a wide community of users and a wonderful sense of history and tradition. As a consequence, we have been able to get "way off the ground" very fast.

We have added to the environment provided by INTERLISP a set of functions which mimic the basic mail processing functions of HERMES. They permit us to analyze message files into internal format. They also include functions to aid in manipulating this internal format. In addition to these functions we have constructed some of the HERMES operations. In particular, it is possible to associate objects such as messages, templates, compose templates, sequences (we call them domains) and filters (we call them selectors) with files. We can transcribe messages through templates, and do

simple composition through compose templates. Selectors include arbitrary logical operations.

We have plans for extending this initial message system in the months to come. In particular we plan to extend the operations which can be invoked from both printing and composing templates to include control primitives such as conditionals, iteration, and indirection. It is our hope that, as a result, it will be possible to create a template which mimics HERMES's reply and forward commands.

The preceding paragraph exemplifies the methodology we are using in this effort: we try to implement, using the facilities extant at some time, to achieve some activity which users of message systems might want and which they might possibly be able to program for themselves. If we have trouble, we study what facilities would allow us to achieve our goal. These are added and new applications are tried. This results in the growth of a language for talking about activites users want to have happen while processing messages. This language embodies the user's level view of the functionality of message processing (as exemplified by the then current version of the system).

5.3.2 The Design of Human Interfaces

We turn now to the design of human interfaces. A system with functionality which supports well the needs of its users stands or falls on how well that functionality is presented to those users. In spite of this fact, designers often treat the front end as an afterthought, tacking on a command structure that roughly mirrors the functions in the back end but adds a few user-oriented touches like argument completion, spelling correction, prompting and some form of on-line documentation.

If the designer turns to the human factors literature for guidance, he finds studies on such issues as function keys vs typewriter keyboards, or fixed argument ordering vs keyword marking of arguments. Even special studies to aid in the design of a specific system focus on issues like the proper names for commands or the keyboard and display layout.

We have come to the conclusion that these types of front-end differences play a very small role in determining the usability and types of interactions a front end will support. This opinion has arisen in part over the efforts of the past few years in providing HERMES with a habitable front end.

We therefore spent some time reflecting on the nature of front ends, and in particular on determining what factors of front end design (in contrast to the keystroke and button-push sort) play important roles in the habitability of those front ends.

The result of this informal study of front end capabilities is an identification of a number of "dimensions" along which front ends can vary. These include: character-at-a-time vs. line-at-a-time; what forms of aid in constructing commands should be available; how are these aids invoked; what is the nature of a command; how does the user make the request that he be shown that information; how is that nature presented to the user; how is ambiguity handled; how is the user informed of it.

To consider interactions between different values of these dimensions, we felt it was necessary to construct a "front end lab" which would permit us to quickly construct front ends exemplifying various points in the "space" defined by these dimensions, thus permitting us to explore the "feeling" of front ends having new combinations of features.

We decided that this lab would consist of a collection of tools which could be used to construct new front ends. The tools would include:

a) A general parser, like an ATN building system that can be easily modified to deal with a broad range of language structures. The parser-building system would include hooks for self-description, assist and adaptability.

b) A history list, with hooks to enable implementation of assist features.

c) A scope interface package. This would handle the display, windows and communications between windows.

d) A communications package. This would handle the communications between a set of back-end processes and the front-end lab. It would allow both synchronous and asynchronous communication and deal with passing whole programs, written by the front-end, as well as sequences of commands.

6. THE LDMX-TENEX INTERFACE AT CINPAC

The translator between the incoming LDMX messages and the messages used by the TENEX System, which conform to the ARPANET standards, was completed, given a final debugging and installed at the CINCPAC site on Oahu, Hawaii.

During this operation, we found errors in the CINCPAC LDMX software, and assisted in correcting them.

We participated in the training of the CINCPAC operator, and furnished a minimal set of operating instructions.

On June 25 and 26, we participated in a final test, which resulted in the acceptance of the translator by the Navy.

The remaining tasks for this project are the completion of a user-interface program and the production of documentation of the complete system.

APPENDIX A
Benchmark Tests of HERMES 4.0.12, MSG and HG

		HERMES 4.0.12		MSG		HG	
		Load Avg.	CPU sec.	Load Avg.	CPU sec.	Load Avg.	CPU sec.
			Real sec.		Real sec.		Real sec.
Overall Time*	10 msgs	--	1:29.5	--	1:20.3	--	55.1
			46:21		25:11		43:16
	500 msgs	--	2:54.7	--	4:57.7	--	57.2
			46:37		1:11:08		26:54
Start-up	10 msgs	5.7	8.7	6.3	4.4	8.2	1.4
			2:12		1:19		0:23
	500 msgs	8.60	8.2	14.21	2:38.7	14.7	19.7
			1:50		30:18		5:33
Quit plus Expunge	10 msgs	--	7.3	21.14	2.3	4.87	1.3
			0:42		0:40		0:25
	500 msgs	26.48	1:42.2	14.03	29.0	17.63	19.2
			19:1		8:56		7:15
(1)Survey From Jones	10 msgs	19.76	1.5	21.84	0.6	21.65	0.4
			0:54		0:22		0:24
	500 msgs	14:62	34.6	4:11	1:47.7	18.8	1.4
			12:06		25:55		2:12
Survey Subj Roses	10 msgs	7:24	1.4	19.07	0.6	19.42	0.4
			0:26		0:11		0:12
(2)Survey From Jones	500 msgs	18.48	3.2	4.68	6.9	11.53	1.3
			1:07		1:37		1:59
Survey From Jones Subj Foo	10 msgs	4.89	1.9	23.09	1.0	10.48	0.8
			0:48		0:28		0:26
	500 msgs	17.48	3.3	4.31	16.1	18.45	1.4
			1:54		2:06		2:32

*The overall time for the 10-msg test is greater than the sum of the individual times. This was a long test and some commands were omitted in the analysis.

Delete 1,3,5,7,9	10 msgs	-- 1.9 0:48	23.48 0.4 0:18	8.03 0.4 0:17
Delete 10:20	500 msgs	11.5 1.3 0:51	2.74 0.4 0:13	18.03 0.5 0:29
Delete 100:200	500 msgs	10.5 4.1 1:05	2.57 0.6 0:15	17.00 2.2 0:29
Delete 450:500	500 msgs	8.4 2.5 0:59	2.43 0.5 0:11	17.30 1.3 0:34
Print 1	10 msgs	20.77 1.9 1:26	22.49 0.9 0:40	23.09 1.0 0:46
Print 1	500 msgs	9.5 4.8 2:24	2.05 1.9 1:45	19.70 1.8 1:35
Print 6	10 msgs	19.76 3.1 1:37	21.62 1.9 1:35	21.14 1.1 0:52
Print 449	500 msgs	12.1 9.8 4:53	1.90 4:3 3:17	18.22 3.9 3:22
Compose 1st	10 msgs	14.39 4.1 3:20	18.53 4.7 1:39	21.07 2.0 2:20
Compose 2nd	10 msgs	19.39 3.3 1:37	27.70 5.4 2:24	24.67 2.1 1:51
Reply 5	10 msgs	28.31 3.9 2:22	18.94 6.7 2:24	19.71 2.9 1:47
Move 3	10 msgs	30.20 2.0 0:26	20.83 0.6 0:29	18.77 0.6 0:37