PAUL BLATTNER
DATA DYNAMICS, INC
RT 2 BOX 502A
PORTLAND OR 97231
Steve Heitmann, President

A BRIEF OVERVIEW of THE AIM SYSTEM
DATA DYNAMICS, INC.

WHAT IS AIM? AIM (Active Information-Management) is a computer-based personal information support tool for professional and clerical workers. It is a machine-independent software package that provides person-to-person communications, document preparation and information management facilities.

WHAT CAN AIM DO? AIM provides a fast, reliable and paperless alternative to conventional methods of preparing, distributing, organizing, storing and retrieving correspondence, documents or publications.

With AIM's document preparation capabilities, short memos, forms or extensive documents can be composed and edited simply. When a paper copy is required, automatic document formatting can be used to prepare it for printing.

Completed correspondence, articles or documents can be distributed using AIM's electronic mail capability. They can then be read, summarized, forwarded, filed or disposed of simply and electronically. Since AIM provides an alternative to most person-to-person communication modalities, it can be used in more diversified applications than a simple electronic mail system. For instance, through the AIM system, asynchronous meetings can be conducted, publications or bulletins can be sent to subscribers, questionaires can be distributed and automatically collected or reminders and scheduling can be used to facilitate many activities - AIM provides communication capabilities that facilitate complex social or business information exchange.

Information management is a vital requirement. Complex or voluminous information becomes much more meaningful and useful when it can be organized and located easily. AIM meets this requirement with its flexible and simple-to-use information management capability. Information can be stored by category, cross-indexed and retrieved by key-word or simple association. Although this capability is intended primarily for use with information generated and exchanged by AIM users, it is also possible to cross-index and retrieve references to paper-based information files.

WHO CAN USE AIM? Prior to 1982, large business and scientific organizations with existing timeshared computing facilities would be the most likely users. After 1982, when inexpensive networked personal computers emerge, most organizations could use AIM. AIM is designed for use on either timeshared or networked computer systems.

As far as user requirements are concerned, AIM is designed to be used by clerical or professional individuals who have no experience with computers. AIM is very easy to learn and use. It is a friendly, "forgiving" system: if necessary information is missing from an instruction, AIM requests the user to provide it. Assistence is readily available. When the user types "HELP", AIM explains what commands can be used and how to use them.

WHAT AIM WILL DO FOR YOU In short, AIM will save you time and money while increasing your effectiveness. Although many more advantages and uses of AIM will certainly emerge with wider use, the following benefits have already been

experienced by users:

1. Conserves resources by reducing the need for paper and office supplies.
2. Reduces copying cost and labor time; reduces copy machine maintenence.
3. Reduces intraorganizational mail costs and the time and labor required to deliver it.
4. Reduces secretarial load by decreasing dictation, retyping, phone calls, etc.
5. Allows communicators a greater amount of time for thought than do face-to-face meetings or telephone conversations.
6. Some face-to-face meetings will be eliminated.
7. Facilitates more organized face-to-face meetings.
8. A wider range of opinions can be evaluated more quickly.
9. Time and distance barriers to communications are removed.
10. Allows longer interruption-free periods in which to work.
11. Routine correspondence can be handled more quickly and efficiently.
12. Facilitates better use of personal time through increased organization.
13. Allows managers to prioritize workload and problems.
14. Insures higher message security than does conventional mail.
15. Can lead to a productivity or performance increase.
16. More employees can be effectively managed without additional managers.
17. Facilitates task assignment and follow-up evaluation.
18. Increased documentation of activities and communications.
19. Questionaires will be more effective.
20. Marked decrease in paper files and personnel time required to maintain them.

AIM AVAILABILITY AIM will be introduced initially as a computer-based person-to-person communication system. This basic system has three enhancements which already exist and will be available during the 15-month period following its initial release: → "SPECIFIED"

Initial version will include (scheduled release: February 1980):

1. Electronic Mail and Computer Conferencing
2. Publications and Bulletin Boards
3. Directory Facilities
4. Form specification and preparation
5. Advanced editing capabilities
6. Elementary filing and retrieval capabilities

Enhancements:

1. Document Preparation (Word Processing)
   Scheduled release: June 1980
2. User-defined programmability using the PASCAL language
   Scheduled release: August 1980
3. Data base management with advanced retrieval capabilities
   Scheduled release: March 1981

INFOSCOPE

ANDREW A. AINES

THE GREAT PENTAGON CAPER. They finally did it at the Pentagon. They dismantled the pneumatic message terminals on the "A" Ring--25 of them--and nobody seemed to notice. It did not matter that they have have not been in use for several decades; it seemed that those stations were there forever. Nobody--certainly not generals and admirals--would have had the courage to order them removed, not even Admiral Hyman Rickover, but someone did. Not a word was printed about the caper in the *Washington Post*. In that wonderful and sentimental *Congressional Record*, which is given to on-beat and offbeat tributes--not a word. Neither Jack Anderson, who keeps a beady eye on the federal beat, nor Senator Proxmire, originator of the Golden Fleece Awards, provided an obituary. There may be a few old-timers around who can remember when the messages went down (and up) the tubes in the Pentagon, but even they were not around to shed a tear. There were no ceremonies, no long-winded speeches, no protesting pneumatic-tube lovers. There was only silence, but some of us have faith. One day, when visiting the Smithsonian, we will come on an exhibit of Pentagon pneumatic message tubes, right next to the exhibit of Alexander Graham Bell's first telephone, just beyond the exhibit of the last newsreel.

FAST-FAC. What is the future of facsimile transmission? INFOSCOPE is bullish on the subject. The great improvement in speed of facsimile transmission virtually guarantees that this medium will be increasingly used in the future. The latest datum for the jigsaw puzzle is an announcement that the Federal Communications Commission (FCC) has authorized Graphnet Systems, Inc. to open a new data-transmission service between the United States and Europe on an experimental basis. There will be facsimile links between data terminals and facsimile devices in 11 Western European countries and the United States. It is too early to determine the annual growth in traffic via facsimile, but it stands to reason that it will increase, especially in the transmission of important commercial and government documents. The hopes are high that other bibliographic and statistical information will be transmitted as costs decrease and speed of transmission increases.

PRESTO, PRESTEL! While the U.S. Postal Service pleads for a chance to get into electronic mail service to keep up with changing communication demands and the need to provide jobs for its 650,000 employees, the British Post Office has started its Prestel information service that makes 146,000 pages of facts available to users through telephone hookups and home television screens. It costs $2000 for a specially adapted TV set, but the cost will be lowered to about $64 by 1983. The West Germans and the Dutch have accepted Prestel, but France and Canada are developing their own systems. The British Post Office is working on the United States as a possible customer for its system, while the U.S. Postal Service is trying to find its way out of a cul-de-sac.

INFORMATION GROUND-UP. A U.S. company has signed up with a French firm to supply federal documents and related research articles to European clients.----The Ayotollah Khomeini praised Islamic workers who seized Iran's largest newspaper, declaring that newspapers "must write only what the people want."----Telenet and ITT World Communications are offering a new service called INFOTEX. Telenet user who travel internationally can access their own host computer systems from almost anywhere in the world, placing a Telex call instead of a telephone call to obtain the service.----Implementing the new wiretap law, Chief Justice Burger named ten judges to serve on wiretap courts. These courts must authorize all national security wiretaps.----Xerox Corp. has entered the low end of the copier market with an inexpensive desk model that makes about ten copies per minute and will rent for about $60 a month.----Famous science-fiction author, Ray Bradbury in a Pertec Computer Corp. ad, says that: "Any computer that can memorize Aristotle *is* Aristotle. Any book that remembers the words spoken by da Vinci is, in a strange, lovely way, da Vinci. A book is a relic of some old words now refound. Computers are a new kind of reliquary for old wise bones as well as dumb."

POLITICAL NEWS. Now that 1980--a big political year--is around the corner, INFOSCOPE cannot resist repeating Sidney Ascher's anecdote, printed in the Pleasantville (N.J.) *Mainland Journal*. When columnist Norton Mockridge was the city editor for a now defunct big-city newspaper, he was questioning a reporter who covered a political rally. "What did the great candidate have to say?" Mockridge asked. "Nothing," said the reporter. "Okay," said Mockridge, "keep it down to one column."

REPORTERS' DELIGHT. With justifiable pride, *Computerworld* claims that it is one of the first publications to "join the electronic mail revolution." In minutes, its reporters can send stories to their editorial office in Newton (Mass.) from any place in North America at a cost lower than overnight Telex service. To make it possible, the Dartmouth Time-Sharing System and the Telenet network are tied together. A 13-lb. TI 745 terminal is used by the reporter. Costly long-distance telephoning is eliminated. Down the road is the TI 765 terminal with bubble-memory storage, which will enable the reporter to compose his story "off-line" and then transmit it to *Computerworld*'s computer at maximum transmission speed. This is splendid, but will the U.S. Postal Service find its way out of the cul-de-sac? ■

CNP-14

BSR X3.57

X3S33/125
Rev 1974 Nov

X3 Project 47

Draft Proposed
American National Standard

MESSAGE HEADING FORMATS

FOR INFORMATION INTERCHANGE

USING THE ASCII FOR DATA COMMUNICATION

SYSTEM CONTROL

This draft standard is published for a four-month period of public review and comment. Comments received during this period will be considered and answered. The draft standard, revised as necessary, will then be submitted to American National Standards Committee X3 for letter ballot. Upon completion of the ballot, the proposal will be forwarded to the American National Standards Institute for approval as an American National Standard.

During preparation of this proposal it was circulated to the several subcommittees of X3. There developed a difference of views between the originating subcommittee X3S3 - Data Communications and X3L5 - Labels and File Structures. They agreed that the differences concerned the goals of the standard, and were not susceptible to resolution by minor technical changes. After review, X3 concluded that this dialog should be circulated with the proposed standard for public review. The correspondence is appended as Attachment A.

Comments should be returned as soon as possible but not later than 1976 April 15, addressed to:

CBEMA/Secretary X3
1828 "L" St NW
Washington DC 20036

Prepared by

Task Group X3S33 - Data Communications Formats
Technical Committee X3S3 - Data Communications

American National Standards Committee
X3 - Computers and Information Processing

Secretariat: Computer and Business Equipment Manufacturers Association

FOREWORD

(This Foreword is not a part of the American National Standard for Message Heading Formats for Information Interchange using the ASCII for Data Communication System Control, (X3.4-1968)).

This standard specifies message heading formats for use in information interchange between systems using the American Standard Code for Information Interchange (ASCII).

Specified in this standard are the definitions of the message heading items, the sequence of these items, the means of indicating which items are present in the heading, and the use of the ASCII delimiter characters to separate the heading items.

Historical and present practices were considered in the development of this standard.

Other standards prescribe the character structure and character parity sense; the bit sequence, the signaling rates; the data link control procedures; error control; and other parameters vital to the communication of information between systems.

This standard was approved as an American Standard by the American National Standards Institute on

Suggestions for improvement gained in the use of this standard will be welcome. They should be sent to the American National Standards Institute, Inc., 1430 Broadway, New York, New York 10018.

TABLE OF CONTENTS

1. SCOPE

1.1 This standard specifies the information items used to construct a message heading and prescribes the sequence of these items. It is intended to satisfy the message heading format requirements for general interchange of information between systems that employ the character set of the American Standard Code for Information Interchange (ASCII; American National Standard X3.4-1968) and the ASCII Data Link Control Procedures (American National Standard X3.28-1971) for data communication system control.

1.2 The primary factor in the design of this standard was to provide the capability of interchanging information between systems, and secondarily, to provide a method of obtaining subsets of the general capability for optional use within a given system.

1.3 Two levels of conformance with this standard are recognized as specified in Section 5, CONFORMANCE.

2. INTRODUCTION

A message is a sequence of characters arranged for the purpose of conveying information from an originator to one or more destinations (addresses). It contains information (called text) to be conveyed from the composer to the recipient(s), and may, in addition, contain supplementary information (called heading).

The need for message heading information varies widely depending upon the applications considered. In some applications, a heading may be required to convey system control

information, such as addressing, identification and status instructions. In other applications, however, no heading information may be required.

This standard defines and positions heading format information items that may be used to perform such functions as:

a. handling and delivery of messages,

b. processing of messages,

c. billing and accounting of messages.

The procedures to perform these functions are not covered in this standard.

3. GENERAL

3.1 Message Sections

A message normally consists of a message heading and a message text. In some cases, a heading is not required. The text of a message contains information that the message originator wishes to be conveyed to the message addressee(s). The heading of a message contains supplementary information that may be needed by the communication system and/or the destination station to handle the message.

The message heading, if used, is associated with a message text and is applicable only to the text that immediately follows it.

In general, the contents of a message are expected to be provided by the message originator. In handling a message, the communication system may not make alterations to the contents of a message heading except as provided for in this standard. The communication system must deliver the contents of the text of a message to the message addressee(s) without alteration. See Appendix F.

3.2 Message Framing

A message is framed by three communication controls (two if a heading is not used). One control delimits the start of the message heading, one delimits the start of the message text and the third control delimits the end of the message text.

Two types of messages are defined: A Basic Message and a Transparent Message. In a Basic Message, any of the 118 non-communication control characters of ASCII may be used by the message originator in the preparation of a message heading (subject to the provisions of Section 4) or a message text. In a Transparent Message, any of the ASCII control characters may be present in the message text. The heading of a Transparent Message, however, is subject to the same 118 non-communication control character restriction established for Basic Messages. (For ASCII data link control procedures, see American National Standard X3.28-1971).

Different framing controls are employed for the two types of

messages. For Basic Messages, the ASCII communication control characters SOH, STX and ETX are used to represent the start-of-heading, start-of-text and end-of-text control functions. For Transparent Messages, the communication control character sequences DLE SOH, DLE STX and DLE ETX are used to represent these functions. For consistency, all following descriptions and examples will use the Basic Message control characters SOH, STX and ETX; to convert to Transparent Message control character sequences, substitute DLE SOH for SOH, DLE STX for STX and DLE ETX for ETX. The two-message formats are shown in Figures 1a and 1b.

Message framing controls are considered to be a part of the message.

3.3 Message Heading

3.3.1 General

A message heading is subdivided into two sections: An Address Section and an optional Reference Section.

The Address Section is started by a start-of-heading control character (SOH) and is ended by a start-of-reference indicator (SOR) which is defined in this standard as the File Separator character (FS) or, in the absence of a Reference Section, by a start-of-text control character (STX). See Figure 2.

The SOR indicator starts the Reference Section of the message heading, which continues until a start-of-text control (STX) is reached.

3.3.2 Message Heading Address Section

The Address Section of the heading contains the information necessary for the communication system to handle the message and to route it to its ultimate destination(s). In some cases, the Address Section may also contain information associated with the transmission of the message over a particular data communication link. Link information in the Address Section may be provided by the originator of the message or by the communication system. When transmitted on a given link, however, a message heading Address Section should contain only that link information applicable to the link in use.

Information in the Address Section, other than link information, should be provided by the message originator. This information may be altered by the communication system (subject to the provisions of Section 4.2) during the process of delivery of the message to its destination(s).

All messages having a heading will have a Heading Address Section, which, as a minimum, will contain the Heading Item Indicator (HII). See Section 4.2.1

3.3.3 Message Heading Reference Section

The Reference Section of the heading contains communication and processing information that is to be delivered with the message to the destination station(s).

In abnormal message delivery circumstances, the Reference Section may be used by the communication system to determine the disposition of the message.

In general, the contents of the Reference Section are to be delivered exactly as provided by the message originator. In certain cases, however, the communication system may add information to the Reference Section (See Section 4.3).

4. MESSAGE HEADING ITEMS

This section prescribes the function and composition of message heading items. The communication system is not permitted to delete these items except as specified in 4.2 and 4.3. Revision and updating are allowed as specified in 4.2 and 4.3.

The use of any message heading item is optional with the exception of the Heading Item Indicator (HII). Message heading items may have variable lengths. Note that the Heading Item Indicator (HII) is a minimum of two characters in length.

Heading items, when included in a message, only pertain to the message of which they are a part.

4.1 Message Heading Item Separators

Four ASCII characters have been defined for use as heading item separators. These characters are the File Separator (FS), Group Separator (GS), Record Separator (RS) and Unit Separator (US). These characters are used to frame heading items and may not be used within any heading field.

The separator which defines the start of a particular heading item is specified in the subsequent description for each item. See Appendix E for description of separator use and examples of message headings.

4.2 Message Heading Address Items

In addition to the Heading Item Indicator, there are six items in the Heading Address Section numbered (HA1) through (HA6) to facilitate identification. The sequence of these message heading items is shown in Figure 3. See also Appendix C.

4.2.1 (HII) Heading Item Indicator

The Heading Item Indicator is preceded by the control character SOH, and is composed of two ASCII characters which identify those items present in the message heading. This item is present in all messages containing a heading. See Appendix D for description of HII bit significance.

4.2.1.1 (HII)Extension

The Heading Item Indicator may optionally be extended where such extension may be useful to furnish additional information regarding the heading or message content, for example to:

a. Furnish a count field (bits, characters, cards, blocks, number of addresses, etc., in the heading or text).

b. Indicate block or field lengths.

4.2.2 (HA1) Link Message Identity/Date-Time Group

HA1 is a unit of information which if present is preceded by the separator GS, and may be used to identify a message on a given communication link. This heading item contains only that link message identity applicable to the link in use. The link message identity may include a date-time group to identify the date and/or time at which the message transmission occurred on a given communication link.

4.2.3 (HA2) Link Message Status

HA2 is a unit of information which if present is preceded by the separator RS, and may be used to convey status information pertaining to the current transmission on a given communication link. This heading item contains only that link message status information applicable to the link in use.

4.2.4 (HA3) Privacy/Classification

HA3 is a unit of information which if present is preceded by

the separator character GS, and is assigned to a message by the message originator. This item indicates the degree of precaution that should be exercised by the system to avoid unauthorized disclosure of the message. When used, the privacy designator is considered to be a permanent part of the message and is to be delivered without alteration by the communication system to all addressed stations.

4.2.5 Destination Address Items

The message heading items HA4, HA5 and HA6 are applicable on a per address basis and if present must be contiguous for each address. For multiple address messages, sequences of items HA4, HA5 and HA6 are used within the Heading Address Section.

For multiple address messages, where all addresses have the same precedence, HA4 may optionally be used only once; but in this case, it must always precede the first HA5 address field. Note that where a single precedence is used on multiple address messages, care must be taken on interchange links to insure that the precedence is forwarded with those associated destination addresses requiring action by the receiving station.

4.2.5.1 (HA4) Precedence Indicator

HA4 is a unit of information which if present is preceded by the separator GS, and designates the degree of urgency for

the delivery of a message to a particular addressee. A precedence indicator, when used, is applicable only to its associated destination address. It should be inserted in the message heading by the message originator. For a multiple address message with a single precedence, see 4.2.5.

4.2.5.2 (HA5) Destination Address

HA5 is a unit of information which if present is preceded by the separator character RS, and is supplied by the message originator. This item identifies the station or stations to which the message is to be delivered. When transmitted on an interchange link, this message heading item contains only those destination addresses (with any associated precedence indicators and secondary routing information) requiring routing action by the receiving system.

4.2.5.3 (HA6) Secondary Routing/Handling Information

HA6 is a unit of information which if present is preceded by the separator character US, and is supplied by the message originator. The content of HA6 is associated with the immediately preceding destination address and, when present, is used to facilitate the handling of a message after it has arrived at the destination station. This information, for example, may include identification and location of individuals, departments, organizations, or devices. This item may also be used by the communication system when that system

has to perform various communication based functions, e.g., code translation, as a part of moving the message through the communication system or delivering it to the addressee.

4.3 Message Heading Reference Items

Heading Reference Items, if used, are a permanent part of the message, and are to be delivered to all addressed stations. Unless designated in the item description, no Heading Reference Item may be altered by the communication system. The sequencing of message heading items in the Heading Reference Section is shown in Figure 4. See also Appendix C.

4.3.1 (SOR) Start-of-Reference Character

The Start-of-Reference Character (File Separator (FS)) delimits the beginning of the Heading Reference Section.

4.3.2 (HR1) Reference Station Identity

HR1 is a unit of information which if present is preceded by the separator character FS. This item identifies the station that performs communication servicing functions for the message originator. The reference station identity is intended principally for use as a means for a message addressee or the system to perform communication servicing functions (e.g., requesting a repeat transmission of the message). In addition, some systems may make use of this information in abnormal circumstances concerning message delivery. The reference

station identity is not necessarily the address of the originator of the message.

4.3.3 (HR2) Originating Station Identity

HR2 is a unit of information which if present is preceded by the separator character GS. This heading item identifies the address of the originating station.

4.3.4 (HR3) Originating Message Identity

HR3 is a unit of information which if present is preceded by the separator character RS. This heading item distinguishes a message from other messages transmitted from the same originating station. The originating message identity need not uniquely identify the message in the absence of additional information. Additional information which may be needed to uniquely identify the message may be implied by the administrative environment (e.g., reference to message 100 might always imply today's message 100) or may be contained in another heading item (e.g., a date-time group).

4.3.5 (HR4) Originating Date-Time Group

HR4 is a unit of information which if present is preceded by the separator character US. This heading item identifies the date and/or time the message was first entered into the communication system. When an originating date-time group is used, it should indicate the date and/or time of transmission by the originating station.

4.3.6 (HR5) Message Accounting Information

HR5 is a unit of information which if present is preceded by the separator character GS. This heading item may consist, for example, of an account or terminal identification, which is to be billed for a message transmission. The information content of HR5 may be modified as necessary by the communication system.

4.3.7 (HR6) Message Status

HR6 is a unit of information which if present is preceded by the separator character GS. This heading item is added to a message by the message originator or by the communication system to indicate the delivery status of the message. For example, a repeat transmission of a message (suspected duplicate), or a message presumed (by the communication system) to be in error (e.g., due to interrupted transmission), may be so designated in the message status item.

5. CONFORMANCE

Two levels of conformance with this standard are recognized.

5.1 Total or Class A Conformance

Total or Class A Conformance with this standard exists when all stipulations of the standard are followed.

5.2 Partial or Class B Conformance

Partial or Class B Conformance with this standard is recognized because systems planned prior to the publication of the

standard may not be able to conform fully, either for economic reasons or because the equipment cannot generate all of the ASCII control characters (e.g., the information separators and all of the binary encodings of HII). Therefore, in order to promote the interchange of traffic between these systems, and also to facilitate interfacing with Class A systems, partial conformance will exist when all stipulations of the standard are followed except that:

a. Use of HII, Heading Item Indicator, is optional and, when used, the first two characters may be redefined.

b. Use of the standard Heading Item Separators is optional and other means not defined herein may be used to identify the presence and/or location of heading items.

S		S		E
O	Heading	T	Text	T
H		X		X

Figure 1a - Basic Message Format

D S		D S	Transparent	D E
L O	Heading	L T	Text	L T
E H		E X		E X

Figure 1b - Transparent Message Format

S	Heading	S	Heading	S
O	Address	O	Reference	T
H	Section	R	Section	X

Figure 2 - Basic Message Heading Format

Separator Character	Item	Description	Reference
SOH		Start-of-Heading Character	
	(HII)	Heading Item Indicator ----------	4.2.1
GS	(HA1)	Link Msg Identity/Date-Time Group ----	4.2.2
RS	(HA2)	Link Message Status ----------	4.2.3
GS	(HA3)	Privacy/Classification ----------	4.2.4
GS	(HA4*)	Precedence Indicator (per address) ---	4.2.5.1
RS	(HA5*)	Destination Address (per address) ----	4.2.5.2
US	(HA6*)	Secondary Routing/Handling Information (per address) ----------	4.2.5.3
		Heading Address Section	

Figure 3

FS		Start-of-Reference Indicator (SOR)	
		File Separator Character (FS) --------	4.3.1
	(HR1)	Reference Station Identity ----------	4.3.2
GS	(HR2)	Originating Station Identity ---------	4.3.3
RS	(HR3)	Originating Msg Identity ----------	4.3.4
US	(HR4)	Originating Date-Time Group ----------	4.3.5
GS	(HR5)	Message Accounting Information -------	4.3.6
GS	(HR6)	Message Status ----------	4.3.7
STX		Start-of-Text Character	
		Heading Reference Section	

Figure 4

* There may be many station addresses and/or group code addresses in a message heading. See 4.2.5, 4.2.5.1 and 4.2.5.2 for details.

APPENDIX A - GLOSSARY

A1. In this standard the following definitions are used:

A1.1 Message Composer. The person, program or machine that first writes, generates, or composes the information (message text) to be conveyed to the recipient.

A1.2 Message Originator. The person, program or machine that writes or generates a message in a format suitable for entry into the data communication system.

A1.3 Originating Station. The location (program or machine) at which the composed message first enters the data communication system.

A1.4 Destination Station. The program or machine that receives the message over the last link of the data communication system.

A1.5 Recipient. The person or program for whom the message is ultimately intended.

A1.6 Node, Communication. The connection point for two or more data communication links.

A1.7 Center Switching. An installation in a data communication system where equipment is used to interconnect communication circuits on a message or circuit switching basis.

A1.8 Communication Link. The physical means of connecting one location to another for the purpose of transmitting and receiving data. (ASNSI Vocabulary for Information Processing X3.12, 1970).

APPENDIX B

Criteria

B1. Introduction

B1.1 This Appendix contains the criteria upon which the message heading format was based. Not all criteria have been entirely satisfied. Some of these criteria conflict with others, and the message heading format specified represents accepted compromises of these divergent criteria.

B1.2 The criteria were drawn primarily from communication aspects of information interchange; however, processing and media recording aspects of information interchange were considered.

B2. Specific Criteria

NOTE: Not mutually consistent and not listed in order of importance.

B2.1 The capability of interchanging information between systems should be provided.

B2.2 A method of obtaining message heading format subsets should be provided.

B2.3 Automatic insertion of format control characters should be provided.

B2.4 A unified method of specifying heading formats for simple and complex systems, independent of characteristics of the data system or the transmission system should be provided.

B2.5 Format overhead should be minimized.

B2.6 The format should cause a minimum of confusion to operating personnel.

B2.7 A simple and accurate means of generating the format should be possible.

B2.8 Message heading formats should minimize hardware and software complexity.

B2.9 The standard should encompass all heading functions.

B2.10 Equipment complexity should be minimized when converting from one format to another.

B2.11 Nothing in the format should cause data link control problems.

B2.12 There should be a simple means of uniquely specifying the format content.

B2.13 The standard should be structured to facilitate derivation of logically related smaller sets, including no heading at all.

B2.14 The standard should provide for easy identification of fields within the heading.

B2.15 Each field should stand by itself and not be dependent upon adjacent or surrounding fields.

B2.16 Heading items should be grouped according to function performed and frequency of usage.

APPENDIX C

Design Considerations

C1. General

C1.1 This standard is intended to satisfy the message heading format requirements for general interchange of information between systems that employ the character set of the American Standard Code for Information Interchange (ASCII) for data communication system control. (This is illustrated in Figure C1). The primary factor in the design of this standard was, therefore, to provide the capability of interchanging information between systems, and secondarily, to provide a method of obtaining subsets of the general capability for optional use within a given system. Provision has also been made to permit the system to perform message heading format control functions on behalf of stations.

Note that the link between the systems (see attached Figure C1) may be point-to-point or multipoint.

C2. Determination of Position of Message Heading Items

C2.1 The message heading items are arranged into two groups:

C2.1.1 The first group contains information necessary for the communication system to handle/route a message to its destination; this information may be altered by the communication system during the process of delivering the message to its destination. This is called the Address Section of the heading.

Information items in the Address Section of the heading may be used for fault recovery across a link, for example, in order to request retransmission or to trace a message across a link.

C2.1.2 The second group contains information intended to be delivered in its original form with the message to the destination; this information may not be deleted by the communication system and, except for unusual circumstances, the system may not add any information. This is called the Reference Section of the heading.

Information items in the Reference Section of the heading may be used to perform message accounting in a communication network or for fault recovery from origin to

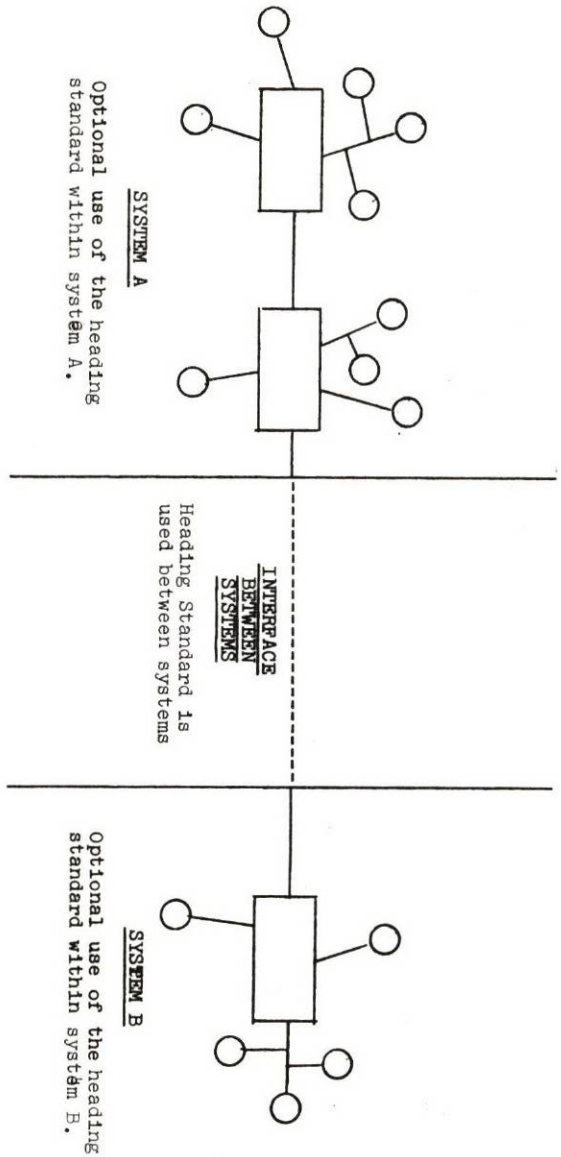


Figure C-1

destination (e.g., to trace or request retransmission of a message).

C3.1 Items within each of the two heading sections should be arranged in the order in which they will be used, keeping in mind the frequency of use of each item:

Address Section

- The first item (HII) describes in detail the contents of the entire heading; it is the index of the heading.
- The link information items (HA1 and 2) are next since the link is the lowest level of control in the system, and each link operates independently of all other links so is likely to have new information to be placed in these items.
- The following item (HA3) is placed ahead of the addressing information since it applies to all destination addresses and therefore need not be repeated for each addressee.
- The next 3 items (HA4, 5 and 6) are arranged as a group to permit more than 1 level of precedence handling and to allow the inclusion of special routing/handling/programming information for each addressee on multiple address messages.

Reference Section

- The first item (HR1) identifies the station to be contacted for communication servicing functions (e.g., requesting a repeat transmission of the message). This was considered to be of primary importance.
- The "originating" heading items (HR2, 3 and 4) (used to identify the originating station and message) were considered the most frequently used Reference Section items.
- The following item (HR5) was considered to be less frequently used than the previous items.
- The final item (HR6) was positioned last to provide a convenient location for the message originator (or the communication system in unusual circumstances) to insert status information.

APPENDIX D

Description of the Significance of Bits in Message Heading Item Indicator (HII)

GENERAL

The Heading Item Indicator (HII) identifies which of the other heading items (HA1 through HR6, except for HA4 and HA6) are present in the message heading, and whether or not the Heading Item Indicator (HII) is extended, i.e., contains more than two characters. Individual bits in the first two ASCII characters of HII indicate the presence or the absence of these heading items, and one bit is used to indicate extension of HII.

Bit b_7 of the first two HII characters is always set to "1" in order to prevent the possibility of creating a control character from columns 0 and 1 of the ASCII Standard Code Table (X3.4-1968).

NOTE: Since $b_7 = 1$, HII may include any of the characters contained in columns 4, 5, 6 and 7. Special care may have to be taken in some systems since the DEL character may occur as the second character of HII. Bits b_1 through b_6 are used to indicate the presence or absence of heading items and whether or not HII is extended.

HII - Heading Item Indicator Composition

The unextended Heading Item Indicator is composed of 2 ASCII characters. Bits b_1, b_2, b_3 and b_5 of the first character indicate the presence (a logical "1") or absence (a logical "0") of the heading items HA1, HA2, HA3 and HA5, respectively, in the Address Section of the heading. Bit b_4 is designated as a reserved bit (always set to "0") available for future designation. Bit b_6 is used to indicate extension of the Heading Item Indicator (HII). Bits b_1 through b_6 of the second character indicate the presence (a logical "1") or absence (a logical "0") of the heading items HR1 through HR6, respectively, in the Reference Section of the Heading. The relationship of HII bit positions to heading items is shown in Figures D1 and D2. (See Appendix E for examples).

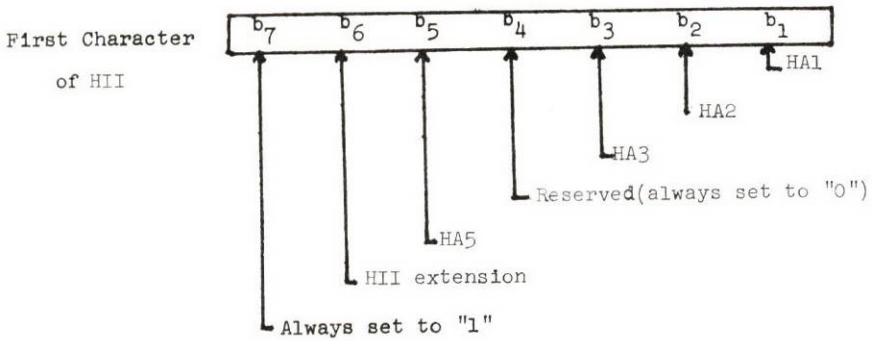


Figure D1

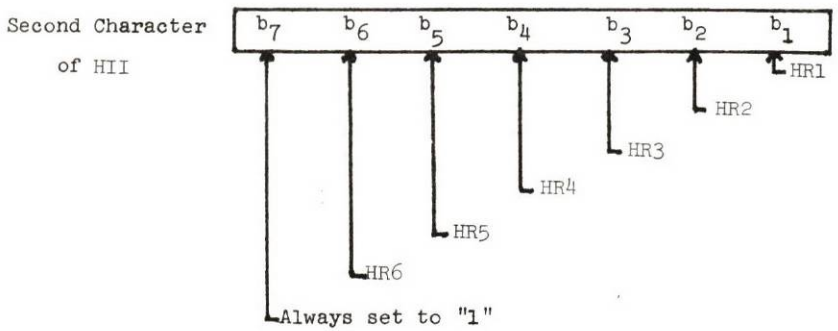


Figure D2

APPENDIX E

Application of Message Heading Item Separators and Heading Item Indicator (HII)

General

The ASCII characters used as heading item separators and the rules governing their usage depart, to some extent, from analogous characters and usage rules employed in systems planned before the issuance of this standard. The purpose of this appendix is to explain the separator character rules and to illustrate how they may be used to form standard message headings of varying types and complexity.

Separator characters are essential in certain types of messages to permit the various heading items and the text to be identified and located easily, especially when these functions are automated. In other types of messages a "fixed" format is used, wherein the size and location of each field in the message is always the same. Still other types of messages contain an indicator (usually at the beginning of the message) that specifies which one of several different "fixed" formats is being used in each message. Typically, systems handling messages with variable field lengths and/or many optional fields have a much greater internal system need for separator characters than do systems handling only rigidly formatted messages. The message formatting needs of different systems are so diverse, however, that even a large number of standard rigid formats would be insufficient to satisfy all of the existing message format needs (to say nothing of future needs). It is for this reason that this standard makes use of separator characters to delineate message heading fields.

In several communication systems planned prior to the publication of this standard, through careful and expert planning, it was possible to combine the functions of field separation with one or more other functions, such as line feed, carriage return, vertical tabulation, horizontal tabulation, form feed and space. In these systems no additional characters were needed solely for indicating field separations because some or all of the above mentioned formatting characters served a dual purpose, thus increasing the overall internal system efficiency. Wherever increased efficiency is attained through dual purpose use of formatting characters, however, restrictions must be made concerning formatting flexibility. For example, if the ASCII LINE FEED character is used to separate message fields, no single field may be longer than one line (unless more complex rules are devised to obviate this restriction). To improve a system's

efficiency, through dual use of formatting characters, the particular rules adopted for field separation and formatting must be custom designed to fit the particular internal formatting needs of the system. Any set of rules involving dual use of formatting characters as field separators will result in format restrictions that are quite undesirable in some systems, though the same format restrictions might have little or no adverse effect on other systems.

Message Heading Item Separators

Considerations such as these have led to the conclusion that heading item separators used for standard message interchange should not restrict the formatting flexibility within any individual systems. This precludes the dual use of formatting characters as heading item separators only for standard inter-system message exchange. It does not curtail the use of formatting characters for any other desired purpose. It was also determined that printing characters are undesirable for use as heading item separators because they could easily confuse or change the meaning of other adjacent printed characters used in message headings. Also, use of any printing characters as heading item separators would prohibit use of those characters for other purposes in message headings - an unnecessary restriction. Four ASCII control characters exist which satisfy the above criteria and they were selected for use as message heading item separators:

UNIT SEPARATOR	US
RECORD SEPARATOR	RS
GROUP SEPARATOR	GS
FILE SEPARATOR	FS

This standard allows the use of these four ASCII characters as desired in the text of a message, but only allows their use to separate heading items in a standard message heading.

In addition to the distinct advantage of not restricting the formatting flexibility within any system, these characters could be inserted automatically into the message heading on the originator's behalf by a terminal device or a

programmed switching center within a given system - prior to transferring the message to a different system. This might be done as part of the routine message processing service provided for the originator in many systems. If, for any reason, use of the standard separator characters within a message heading is found to be undesirable in a given system - and yet that system needs to communicate externally in a standard manner - these characters could be inserted, removed or translated as necessary at the external system interfaces. If this not feasible, partial conformance may be adopted. This option is described in Section 5 of this standard.

Heading Item Indicator (HII)

Used in conjunction with the heading separator characters is the Heading Item Indicator (HII). The first two characters of HII must appear immediately after the Start-of-Heading Character (SOH). When HII is extended the remaining characters and its total length are not specified. One of the fourteen bits in the first two ASCII characters of HII is used to indicate that HII is extended (has more than 2 characters). The high order bit of each character is always set at 1 to avoid generating ASCII communication control characters. One bit is a spare, and the other ten bits are used on a one-to-one basis to indicate the presence or absence of ten of the twelve other heading items (HA1, HA2, HA3, HA5 and HR1 through HR6). No bits in HII are dedicated for indicating the presence of HA4 nor HA6, because these heading items may be present for only some of the addresses in a multiple address message. Thus, HII avoids ambiguities that could otherwise exist where multiple address messages do not always include HA4 and HA6 for each address. It also facilitates automated heading analysis. The three heading items that may appear more than once in a message (HA4, HA5 and HA6) are uniquely distinguishable through their separators.

Heading Item Indicator separator characters may be inserted in a message directly by the originator, or they may be inserted on his behalf by the system. The second alternative may be implemented either through use of an appropriate terminal device at the originating station or by other automated means in the originating system. Studies have shown that control information directly inserted by humans is more apt to contain errors than when automatically inserted. The number of different characters needed for use in the first two character positions of HII (64 characters, whenever the spare bit is assigned for use) necessitated use of printable ASCII characters. In some cases, delivery of the two HII characters to an addressed station or system

may be useful, but in other cases it may not be. In cases where it would not be useful to deliver these characters, an agreement may be made for the transmitting communication system to omit the characters before delivering the message.

Message Heading Format

Examples

The following three examples illustrate use of the Heading Item Indicator (HII) and the various item separators. Example 1 illustrates a "processor to processor" message which would be handled completely automatically. Examples 2 and 3 include forms control characters to aid human recognition. Example 2 includes each of the message heading items defined in the proposed standard. Example 3 includes an extended Heading Item Indicator (HII), two addresses, their secondary routing information and the text. All of the ASCII characters in the message headings are shown; the characters in Examples 2 and 3 are positioned as they might be on a hard copy printout. Forms control characters are omitted from the text portion of the examples. Note in examples 2 and 3 that extra characters may be required at the end of each line for timing purposes to accommodate some unbuffered terminal devices.

Example 1 - HII = SB. When the bits of the first character of HII are 1010011, the ASCII character S, it indicates items HA1, HA2 and HA5 are present, and HA3 is absent, and that HII is not extended.

When the bits of the second character of HII are 1000010, the ASCII character B, it indicates item HR2 only is present (see Figure E1).

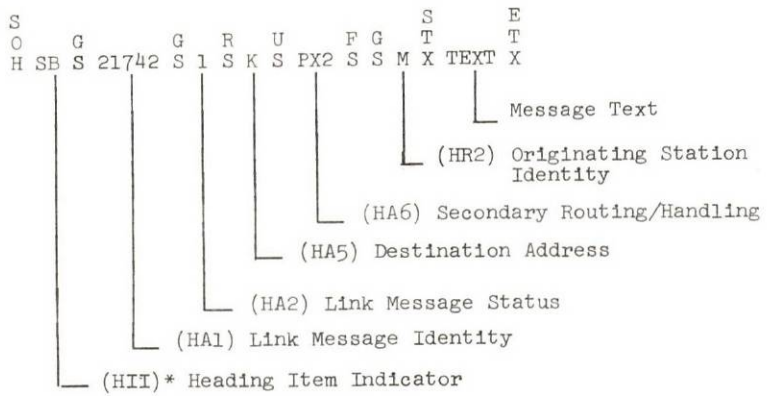
Example 2 - HII = W DEL. When the bits of the first character of HII are 1010111, the ASCII character W, it indicates that Address Section Items HA1, HA2, HA3 and HA5 are present and that HII is not extended.

When the bits of the second character of HII are 1111111, the ASCII character DEL, it indicates that Reference Section Items HR1 through HR6 are present. Note that the message text in Figure E2 is shown garbled to illustrate a usage of HR6. (See Figure E2).

Example 3 - HII = p@02. When the bits of the first character of HII are 1110000 the ASCII character p, it indicates the presence of item HA5, the absence of items HA1,

HA2 and HA3 and that HII is extended. The ASCII characters 02 comprise the extension of HII and in this example are used to indicate the number of addresses contained.

When the bits of the second character of HA1 are 1000000, the ASCII character @, it indicates that no items are contained in the Reference Section; i.e., there is no Reference Section in the heading (and also, therefore, no need to include an SOR indicator which would precede the Reference Section). (See Figure E3).



* Bits 1 through 6 in the first two characters following the Start-of-Heading character indicate which heading items are present. See Appendix D.

NOTE: In this and the following examples spaces around non-printing control characters are added for clarity and readibility.

Figure E1


```
          S   D
          O   E C L
(HII)*    H W L R F

          G       S       S           C L
(HA1)     S MSG. P NO. P R3-1/2/3 R F

          R           S           S           C L
(HA2)     S POSSIBLE P DUPLICATE P MESSAGE R F

          G       S       S       C L
(HA3)     S PVCY P CLASS P 1.7 R F

          G       S       C L
(HA4)     S PREC. P 1.3 R F

          R       S S S L
(HA5)     S NYC P P P F

                  U       S       S       S               C L
(HA6)             S MR. P K. P L. P BROWN(PRES.) R F

          F       C L
(HR1)     S NYC R F

          G       C L
(HR2)     S CHI R F

          R       S       S       C L
(HR3)     S MSG. P NO. P S1 R F

          U       C L
(HR4)     S 1/2/3 R F

          G       S   S       S       S       C L
(HR5)     S BROWN P & P CO. P ACCT. P 3135 R F

          G                               S       S       S               C L
(HR6)     S ***ATTENTION***---THIS P MSG P WAS P INTERRUPTED R F

                    S               S       S               S       C L
          DURING P TRANSMISSION P FROM P CHI---REPEAT P COPY R F

              S       C L
          MAY P FOLLOW R F

          S                                                           E
          T                                                           T
(TEXT)    X TRAVEL REQUEST #39065 REQUIRES YOUR APPROVQZT3 X
```

* Bits 1 through 6 in the first two characters following the start-of-heading character indicate which heading items are present. See Appendix D.

Figure E2

```
        S
        O       C F
(HII)*  H p@02 R F

        R     S S S L
(HA5)   S CHI P P P F

                  U H     S    S       C L
(HA6)             S T MS. P F. P SMITH R F

        H                   S    S       C L
        T              ROOM P 2.3 P (TRAVEL) R F

        R     S S S L
(HA5)   S NYC P P P F

                  U H     S    S       C L
(HA6)             S T MR. P M. P JONES R F

        H                   S    S       C V
        T              ROOM P 3.5 P (SUPPLY) R T

        S
        T
(TEXT)  X TRAVEL DEPT.: PLEASE EXPEDITE TRAVEL REQUEST

          #39065 FOR MR. JONES OF OUR SUPPLY DEPT.

                    K. L. BROWN, PRESIDENT



        E
        T
        X
```

* Bits 1 through 6 in the first two characters following the Start-of-Heading character indicate which heading items are present. See Appendix D

Figure E3

APPENDIX F

Insertion and Revision of Message Heading Information

Prior Agreements

As a message traverses a data communication system the content of the heading portion of the message may be modified from link to link by the system to conform to the agreed upon message structure for each link. For example, a message originator could, by prior agreement, delegate to a switching center or node of his communication system the responsibility of entering certain heading information items on his behalf, such as the Heading Item Indicator, Reference Station Identity, Originating Message Identity, etc. Such delegation would result in a subsetted message heading format on the link from the message originator to the first switching center or node; however, the switching center would insert the "delegated" information into the message heading prior to forwarding the message to a general information interchange machine or program that expects a standard heading format.

In the same manner the message recipient or destination station may, by prior agreement, delegate to the last switching center or node of the communication system the responsibility of removing certain heading information items on his behalf. For example, the receiving station may delegate to its switching center the function of removing heading items such as the Heading Item Indicator, Precedence Indicator and Reference Station Identity, if these items are not needed at the receiving station.

Revision of Message Heading Information

In general, the less complex a system is, the less complex will be its internal message heading format needs. Some simple point-to-point systems, for example, will require no message heading information. As the number of stations in a community of interest increases, and as the complexity of the communication system increases, the need for more complex message heading information increases. The most stringent message heading requirements exist when entirely different systems having different internal needs find that they must communicate with each other. This standard is primarily designed to resolve this problem. It is recognized that application of this standard internally within some systems may also result in overall benefits accrued from uniformity of procedures, programs and/or hardware.

In message interchange between two different systems, the exposure to lost, garbled or misdirected messages can be even more serious than within a single system - yet, preventing and resolving such problems is more complex in a multi-system environment. For these reasons, capabilities are provided in this standard to permit the routing of undeliverable messages to control points (Reference Stations), when confusion arises. The capability is also provided for message identification and accountability on both a network and a "per-link" basis. If desired, communication systems using this standard for information interchange may insert Link Message Identity/ Date-Time Group and Link Message Status Information into the message heading to minimize the probability of uncorrectable errors occurring. Some systems may wish to log these message heading items at switching centers to permit tracing the exact route taken in the delivery of any given message back to the message originator or, perhaps, from originator to destination station.

Other heading items that may be revised by the communication system, while a message is en route, include Message Accounting Information and Message Status. These items are intended for delivery to all addresses.

Each switching center in the communication system before forwarding a multiple address message must either:

(1) Delete message heading items HA4, HA5 and HA6 for which it has delivery responsibility, or

(2) By prior agreement, have some other form of identifying these address items in order to prevent, for example, multiple deliveries of a message to an addressee.

EXPOSITORY REMARKS

BRIEF HISTORY

In 1965, Task Group 3, Data Communication Formats, of ANSI Subcommittee X3S3, Data Communication, was directed to conduct a study of message heading formats used for information interchange. The result of this study was the working paper "Heading Formats for Data Transmission (A USASI) Tutorial," which appeared in the June 1968 issue of "Communications of the ACM." Based on that paper and the many helpful suggestions that it stimulated, the decision was made that standardization of message heading formats for information interchange using the ASCII would benefit all parties involved. Accordingly, Task Group 3 was directed in late 1968 to develop a proposed message heading standard for information interchange using the ASCII.

During the next 3 years, through extensive study of existing data processing and communication systems, and review of new systems being planned, Task Group 3 developed the detailed set of criteria upon which this standard is based. These criteria, listed in Appendix B, reflect a balance between the needs of the data communication users, the manufacturers and the operating agencies. Central in every decision made throughout the development of this standard, therefore, was consideration of the criteria, weighing the arguments against each other.

TECHNICAL DEVELOPMENT

1. The main technical issue encountered was whether to develop a message heading standard for information interchange within a data system, as well as information interchange between systems, or to limit the standard to information interchange between systems. The Task Group, after extensive consideration and debate, decided that the message heading standard should be designed for the inter-system application, with the secondary intent that it be applied to the intra-system application when advantageous to the user to do so. Data systems designed primarily for internal data collection, data enquiry and manufacturing control, for example, would have little or no use for such a standard since they often have fixed formats and use restricted code sets in order to attain maximum efficiency. However, if such information is to be exchanged between two or more of these systems, then the message heading standard applies.

2. Another issue was whether to have a series of message heading standards or whether to design the standard to

meet the most stringent requirements, with less stringent requirements being satisfied by subsets of the full capability. The Task Group decided that the compatibility provided by subsets should be the overriding factor. This will allow users to upgrade systems and still be compatible without expensive outlays for new equipment. It will also allow smaller systems to easily interface with general purpose systems.

3. Two means of promoting message heading format compatibility between simple and complex systems were: (1) the flexibility built into the Heading Item Indicator (HII); and (2), the hierarchical use of the ASCII separator characters FS, GS, RS and US.

At the earliest possible point in a message heading, the Heading Item Indicator specified those heading items that are present in a message. These separator characters are used to define the starting point of heading items and, in conjunction with the Heading Item Indicator, uniquely indicate which heading item follows another.

These two features provide flexibility for the user in designing a message heading to meet the system requirements and to facilitate interconnection of simple and more complex systems.

A CMA Conference Highlight!

A Look at the Impact of Bell's Advanced Communications Service

By Howard Anderson
President
The Yankee Group

AT&T submitted to the FCC on July 10, 1978 a petition for authorization to operate the Advanced Communications Service (ACS), *a switched data communication service*, to be provided over their digital, as well as analog facilities.

AT&T said it was of the opinion that under construction and operating authorizations granted by the FCC for Dataphone Digital Service (DDS) and Dataphone Switched Digital Service (DSDS), they are permitted to use these digital facilities for the new ACS service. However, AT&T asks for FCC concurrence to its own conclusion in order to remove all uncertainties. AT&T asks specifically for a *declaratory ruling* to remove any restrictions that might exist limiting the use of the digital facilities for the provision of ACS.

AT&T is changing from the "Bell Data Network" to "Advanced Communications Service" to accommodate the Independent telephone companies. AT&T sees the Independents as "retailers" of ACS . . . much like they are for Long Lines services.

Perhaps the FCC will require a long time to analyze the pros and cons of the new service proposed by AT&T. It has called for comments particularly in view of its possible overlap with data processing. There is no doubt that there will be controversial issues raised by the ACS petition. It is the opinion of The Yankee Group that the FCC will reject the petition and request a full 214 application—which it will probably approve much more quickly than expected, certainly more quickly than DDS. I would bet on approval in 15 to 18 months.

Different sectors of the data communication industry will be impacted in different ways and the one group to gain most will be the users.

ACS is simply a packet switched network . . . a value-added network just like Tymnet or Telenet. Others are Graphnet, ITT Com-Pak (now under development), and the Xerox and Exxon networks, both also now being developed.

Advanced Communications Service is a unique concept for the Bell System to offer; it provides not only nationwide transmission networking, but communications processing and network management services as well.

While Advanced Communications Service does provide the small and medium sized user with the opportunity to use data communications, it is *specifically geared for the large company.*

Our Research Report, "The Impact of AT&T's Advanced Communications Service", states, "Today, 80 percent of the total dollars spent in data communications is spent by only 750 companies. While no company's position is quite as bad as that outlined in the ACS petition, too many companies have separate networks for each and every application—General Motors has 29 different and distinct major networks, Ford has six different networks emanating from the same building. Data communications today is in a state, which, at best, can only be called 'chaos'. In addition to a wide variety of terminals which are incompatible, line utilization levels are poor—the average data communications line today only uses four to six percent of its channel carrying capacity to ship actual data. Network support is nonexistent."

AT&T has correctly recognized that an opportunity exists because today each application demands its own network and, usually, its own terminal. Most of these terminals are inextricably linked to an address and a channel on the mainframe and tied like a rock to that one application.

By providing compatibility between previously incompatible terminals and hosts, AT&T opens up an *enormous new market*, both for itself and for its users. And by providing access between virtual subnets (private data communications through X.75 and X.25) it accommodated all the users who have begun down the path of intelligent and transparent networks on their own.

By providing a "superset" data network, AT&T has added an entirely new dimension to Data Communication, it has opened a truly new market: intercompany communications. In short, the *data equivalent to its switched voice network.*

Applications? Electronic Mail. Intercompany systems of paperless buying. Faster turnaround time. Lower inventory levels. Speedier customer response. The important feature of Advanced Communications Service is that it must and will be tailored to each industry.

Vanilla Service Won't Hack It

"Vanilla" service just won't hack it. Our research goes into considerable length describing specific applications—and specific implementation problems. Drug wholesalers, for example, have different problems than truckers, and manufacturers have different requirements than regional bankers tying into their correspondents software.

It is very clear that the new marketing organization under Archie McGill in AT&T's General Department has done some homework. They have determined from market research in the most important industry sectors the needs of data communication users from which the basic ACS features have been selected. AT&T estimates from this research that by 1983 there will be a total of 3.6 Million terminals and computers operating in the United States, representing a large number of makes, models, and applications. Because of the projected rapid growth of distributed data processing (DDP) the growth of terminals and small computers is generally projected to be between 20 to 30 percent per year so that the number may easily reach 15 to 18 Million by 1990. AT&T projects connecting to 137,000 of the terminals or a (small) four percent.

ACS is intended to satisfy the data communications needs of a wide variety of users as a public service comparable to telephone service. In order to accomplish this, it has to interface with a wide variety of data processing and data communication hardware and software (protocols). As a result there will be almost complete freedom of interconnection, provided there is ACS support for the particular interface used. With 15 to 18 Million terminals and computers projected for 1990, the potential market for ACS subscribers is substantial and is expected to be comparable to the number of main business stations (telephone), which is now (1978) approximately 14 Million and projected to be around 21 Million in 1990. From these projections it is quite clear that the potential market for ACS begins to be of comparable magnitude to the business telephone market, in addition to any private (dedicated) line market for analog (voice grade, TV, broadband) and digital (DDS) lines. The stakes are significant and AT&T is playing to get the dominant share of the total market. It is not unreasonable to expect that the AT&T share of the intelligent packet switch market may be comparable to that of the telephone market, which is now 82 percent.

As a public shared data communications service, ACS will have the flexibility to accommodate a broad range of data communications users and applications. Presently used terminals will be permitted to serve "new applications" and upgrading of terminals without impacting application programs will be possible. Furthermore, ACS is designed to allow "improvements in network technology" without impacting user application programs and terminals. The flexibility of ACS will be available through options of specific features including customized features determined by the needs of a particular user application.

An alternative mode of data communication would be "the Standard Message Feature" which will include message switching, data entry, and remote batch operation. Message transmission is unidirectional and may be directed to single or multiple destinations. Message editing is performed by the terminal operator who may use the "message storage area" of the ACS network. The terminal operator may select the priority for delivery of the message to the "message arrival area" associated with the recipient of the message. There are three modes of delivery available from the Message Arrival Area . . . automatic, scheduled or demand. Verification of delivery of a message will be available.

ACS provides the customer with a great deal of control over his data communications operations by enabling him to define his own virtual subnetwork.

The most important feature of the ACS architecture from the user's point of view is undoubtedly the *complete freedom of mixing all classes of terminals* and all sizes of computers for different applications into his subnetwork. Thus a host CPU for Data Base Access, another for Order Entry Processing, and still another for Materials Control can be all connected to ACS and so can terminals for inquiry/response, data entry, printing, interactive applications, remote job entry, insurance agents' transactions, travel reservations, and so on.

The ACS network will be initially implemented by means of nodes, which control access, provide data switching, routing, and message management (priorities), interconnected by 56 kb/s trunks using existing digital or analog Bell System facilities. Each node will be connected to every other node by at least two disjoint paths. As the network grows, another level of nodes will be added to the hierarchy by geographical regions to perform tandem switching functions.

All nodal functions will require new hardware and software non-existent in the Bell System now. In addition, there will be ACS interface equipment at the customer premises performing the functions of the Channel Service Unit (CSU) and Data Service Unit (DSU) in the present DDS service, plus, very likely, additional functions specific to ACS. However, existing facilities of DDS will be used to the maximum extent possible.

The CSU and DSU provide the capability for testing customer access lines remotely from the telephone company's remote test center. DSDS which was filed in March, 1976, approved by the FCC in June, 1977, together with the extension of DDS services to 64 cities (from 24), would also use the DDS facilities except that it would operate only at 56 Kb/s. However, in view of the ACS announcement, it is doubtful that the DSDS service will ever be implemented on a nationwide scale.

Even today there has been zero DSDS volume and only $20 million DDS volume. DDS will now diminish. But it was great . . . it helped to kill Datran!

As mentioned before, end-users are to gain most from ACS operations and here are the reasons why:

- Users, particularly smaller companies, will benefit from economies of scale by sharing facilities of a public data service.
- Users can form their virtual subnetworks, operationally equivalent to company operated systems, with full control of access but not burdened by capital investment, operating and maintenance responsibilities.
- Users can mix existing terminals and computer hardware because of protocol transparency of ACS.
- Users can expect ultimately lower cost, but particularly lower start-up cost for small users of data communication services.
- It is not clear that AT&T will allow other communications networks to be connected (X.75) from other carriers. This, we feel, is one of AT&T's bargaining chips, although it would be an advantage to a user to have such interconnectability.

The use of [illegible] voice grade lines for data communication will continue to be attractive to the occasional, unsophisticated user, which will most likely include the future "personal computer users". Although there are no ACS tariffs established, it will be very hard to imagine that ACS can beat the ubiquitous telephone for low usage.

DDS service will continue to attract the large corporate user and heavy users such as airlines, although ACS will have some impact, depending on how ACS tariffs will accommodate batch users, and other heavy usage of dedicated data lines. It is expected, however, that DDS and all competing private line services will remain a viable business supplementing ACS. Telephone companies and special common carriers who have now revenues from dedicated private lines used for data communication, will not be seriously impacted by the ACS service. The exception will be Western Union and the SCC's which provide more diversified and sophisticated data communication services, such as Southern Pacific Communications (SPC) operating the former DATRAN switched data services, and SBS, whenever it opens services. The magnitude of the impact, and the applications prone to be impacted by ACS can be analyzed in more detail only after the tariffs are known.

However, this is a dynamic market area and it can be expected that presently provided data services will change tariffs in the future as a reaction to ACS. Furthermore, there is enough growth in private data line services, which recorded a growth of 26 percent per year since 1974, as reported by leased line revenues collected by AT&T.

VANs Will Suffer Most

The highest impact will be felt by the Value Added Networks such as Telenet and Tymnet, who provide packet switched data services. ACS will be in direct competition and will have the added advantage of economies of scale, and more plentiful resources for developing more market areas through more diversified services.

SBS, although not a VAN, will likewise be confronted directly by ACS and will not have satellite facilities to their advantage since AT&T, most likely in conjunction with GT&E, will also use satellite facilities as needed.

SBS is holding "prayer meetings" and is asking users to form "task force" groups (at the user's expense!) to figure out how a company can use SBS!

All companies, whether common carrier or not, providing switched data services in the United States, now or in the near future will have to develop a *survival strategy.*

Telecommunications 12(11):31-54, Nov. 1978

datacom

[redacted], this is from Telecommunication Magazine this month FYI

Tony

What is Electronic Mail...

and where does it fit into the office?

HOWARD ANDERSON
The Yankee Group
Cambridge, MA

Following an introduction by Mr. Anderson, we present an invited four-part story on Electronic Mail, based upon presentations in a symposium by The Yankee Group, this Summer, in several cities in the United States.

"Electronic Mail is a system of communication whereby a human at one end sends a message to a human at the other end. . . . . some or all of which is handled electronically" —the Gospel according to The Yankee Group.

Electronic Mail is in danger of being relegated to the buzzword Hall of Fame because so many companies are attempting to get on the bandwagon. Perhaps it's easier to discuss what electronic mail is *not*. Electronic Mail is *not* a computer-based system whereby remote terminals query a centralized data base on the status of insurance policies; Electronic Mail is *not* even a facsimile system whereby remote warehouses call in their stocking positions for keypunching these data into a computer.

These two examples are certainly *electronic* — but they aren't *electronic mail*. The question arises — is TWX/Telex electronic mail? Mailgram? ATT's Advanced Communications Service? The answer is definitely *yes*. TWX/Telex is electronic and it does provide a system of communications from human to human. Mailgram is exactly the same — except that the final step in the transmission process is handled by letter carriers. But our definition says ". . . some or all of which is handled electronically."

One of the newest forms of Electronic Mail is the ECOM system which the USPS offers — in effect, cheap Mailgrams — and they too are electronic mail.

Electronic Mail need not be hardcopy or even alphanumerics. A new company in Dallas — Electronic Communication Systems — has begun development on a verbal (oral) Electronic Mail system, where the messages are *never* keystroked or prepared graphically. When people talk about the "paperless office" you can't get any more paperless than that!

There are essentially four variations* of Electronic Mail:

1. Carrier-Based Systems and Public Postal Services: There is some overlap here. The Advanced Communications Service from AT&T is really a computer-based message system and Western Union provides public teletypewriter service. In Europe, the British GPO is promoting Viewdata. Satellite Business Systems is providing the bandwidth for Electronic Mail but it is not providing the hardware — the user must do his own integration;
2. Facsimile: Most likely in widest use today;**
3. Communicating Word Processing: While little used, CWP represents a migration towards interconnecting America's offices, which both users and vendors get excited about;
4. Personalized Computer Based Message Systems: The Cadillac of electronic mail systems. Perhaps the only form which really increases office efficiency and makes people more productive. Very expensive.

In this series of articles, we touch upon these variations.

---

* A fifth variation — Private or Public Teletypewriter — is not treated here since, although widely used, we believe it may become obsolete. Howard Anderson

**"One can foresee that within the next two decades, facsimile will be home-affordable, available to the individual consumer and smallest entrepreneur. In fact, facsimile will eventually end up as a part of a global communications utility. This does not mean that all postal service functions will cease. Bulk mail, packages, and communications which are not time sensitive will still require postal service. There will be complementary functions between "electronic mail" and regular mail. That accommodation will be maintained."
Denis Krusos, President, Panafax Corp.

The USPS and Electronic Message Systems (EMSS)

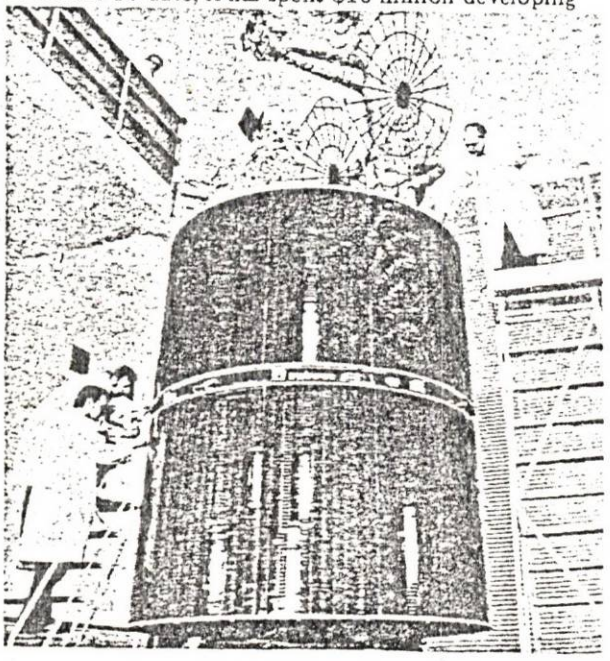
We've all heard about, and most of us have experienced firsthand, the egregious inefficiencies of the United States Postal Service (USPS). And now the cost of mail has risen again — for the fifth time in the last decade — jumping an average of 25% on *all* classes of mail. This boost is no laughing matter for businesses, particularly those using second-class postage, which is climbing 29%.

And will we be getting better service for our bucks? No way. In fact, there's still some talk about cutting back on services. Indeed, since it was "established" in 1970, the government-owned "corporation", known as "The Postal Service" (which was supposed to earn its own way), has raised its rates by over 125%. . .and reduced its services! Nevertheless, the Postal Service still lost 7.5¢ for each piece of mail it handled in 1977 (up from 5¢ in 1974).

One reason for the rise in cost/piece of mail has been the relatively flat growth rate in the *volume* of mail (from 90 billion pieces in 1974 to 92 billion in 1977). This can, in part, be attributed to the rising costs and the decreased services provided by the USPS which has resulted in some companies using private mail-delivery services, others relying more on TWX/Telex and various forms of *electronic communications* (facsimile, communicating word processors) and telecommunications; the citizenry, in general, is sending less personal mail; many businesses and people are simply cutting back.

Indeed, as the USPS continues to flail itself into obsolescence, electronic mail becomes more attractive as an alternative. For one thing, consider that labor costs are rising 8%-10% a year, while electronics costs are falling at 22% a year. The Postal Service, with over a half million employees who have a very strong union behind them, is completely bound by the limits of physical distribution. Some 86% of the USPS costs are directly labor-related. Only 7% are transportation-related.

But the electronics versus labor equation hasn't been completely lost on the USPS. During the last few years, the Postal Service has been investigating electronic message services, for both domestic and international transmissions. To date, it has spent $10 million developing and evaluating the "Concept Phase" of domestic service. This is the first of five phases, which will run through the 1980's.



Intelsat IV-A Satellite.

INTERNATIONAL EMSS

For international electronic messages, the Postal Service recently signed an agreement with Communications Satellite Corporation (COMSAT) to develop a one month field trial (similar to SBS's "Project Prelude") beginning in February, 1979. The Intelsat IV-A Satellite will be used for both high- and low-speed facsimile and OCR overseas transmissions. If all goes according to plan, a one-year field trial will be initiated.

The US earth station will be transmitting from Etom, West Virginia, to a selected group of the 101 countries that are part of Intelsat (the test will involve up to seven countries in Europe, Asia, and South America). The message path will utilize terrestrial links, leased from common carriers, between New York and Washington Post Office "nodes", and the Etam facility. Physical delivery of messages (by users to the Post Offices, and by couriers *from* the Post Offices) will facilitate mail *circulation* through the nodes.

System Planning

One of the most important criteria in developing this electronic message service will be the identification of user groups. Preliminary indications are that users will include: government agencies, banks, international corporations, monetary organizations, and import/export concerns.

Service definition is another important part of the planning phase. How much will it cost? What sort of time guarantees will be instituted? Will there be *levels* of service? How many?

And, of course, the equipment specifications also have to be developed based on the types of services and cost parameters that are desired. The questions here are: Should the service use existing technology, even if it means lowering some performance objectives? Should the Postal Service develop, or subsidize development of new systems?

The upcoming field trial will rely primarily on existing technologies. It will serve to verify the service performance, and to act as a demonstration for user groups. The trial will also aid in cost and operational planning, as well as in implementation. The size of the initial "letter contract" between COMSAT and the USPS is $350,000; the parties are now working out the details of a "definitive contract" in the vicinity of $900,000.

Response

The response to the USPS International Electronic Message initiative has been mixed. While most observers hail the effort as innovative, useful, and crucial to the future viability of the USPS, others claim it is just another government boondoggle, and that the US taxpayer will ultimately foot a large, unwarranted bill. From a different angle, the International Record Carriers (IRC's) are concerned about COMSAT's role in all of this, particularly since COMSAT is now the sole-source vendor. COMSAT is also serving as an advisor for the program — is recommending fax equipment for the field trial.

However, the USPS insists that this will change after the field trial. The USPS will be taking over the entire project management, and it then intends to purchase lines, on a competitive basis, from all qualified IRC's. The FCC will keep a close eye on developments, and will

34 ▶

undoubtedly throw up a few roadblocks just to keep its legal staff busy.

The field test will be the foundation for an international electronic mail service, and is not structured to "back-in" to a domestic service. Nonetheless, it could be interfaced with a domestic service in the future. One possibility is that rather than physically delivering fax messages by conventional means, they could travel directly to (or from) the user's terminal. In fact, it is conceivable that the system's transmissions centers could eventually act as "switching nodes" for incoming and outgoing transmissions.

The *rough* cost estimate for the International Electronic Message Service presently ranges from $1-$4 per page. The goal is to be able to offer "next-day delivery service" at a reasonable cost, as well as lesser levels of service for less cost. It currently costs about $15 for "next day" delivery from the Eastern US to London.

This international electronic message service experiment should serve as a building block for future, expanded services, such as terminal-terminal transmission (as opposed to physical distribution) on the ends. USPS officials, however, insist that it is *not* structured as a "back door approach" to domestic service.

DOMESTIC EMSS

The goal of any domestic electronic message service is next day delivery to 95% of the US, according to John Wise, Assistant Postmaster General for Research and Development. Target date for true operating capability for a domestic EMS system is 1985.

Requirements for a domestic electronic message service include:

- A potentially favorable financial return — absolutely no more congressional subsidies;
- No competition with private sector elements willing and able to serve public needs. Cooperative efforts, however, are acceptable;
- It must be consistent with the broad national policy for information systems (non-existent as yet);
- The substantial resources required must *not* be unduly borne by today's rate payer.

Approach

The USPS is desperately trying to apply the new, automated technologies — computers, electronic transmission, and I/O disciplines — to somehow mitigate the severe limitations of its physical distribution system, while concomitantly providing faster, cheaper, more efficient message/information transfer. Despite the fact that 60% of first class mail stays within 200 miles of its origin, each piece is handled at least a half-dozen times by different USPS employees.

Indeed, the USPS has already experimented with electronic message services — running a "fax mail" experiment in six cities in 1974. Results of that program were mixed, and the service melted into the background. Currently, the USPS is in a "concept study phase" to expand the "fax mail" experiment to 40 cities.

ECOM FAX

Also, the USPS is planning to establish an Electronic Computer Originated Message Service (ECOM). The network will extend across the contiguous US with delivery provided in all 50 states. ECOM will provide USPS selected customers that meet *volume* requirements, and have the necessary *computer capability*, a highly reliable, nationwide service standard of 95% two-day delivery. The USPS proposes to provide ECOM to its selected customers by using a single contractor who will supply the communications *switching* and *transmission* segment of the service.

ECOM messages in non-hard copy form (magnetic tape or disk, intelligent terminals, word processors, etc.) will be accepted by the contractor from USPS customers for transmission via the contractor's communications network to 25 serving post offices (SPO's). The 25 SPO's have been selected on the basis of postal mail processing abilities to service the ZIP Code areas within two postal delivery days of receipt by the SPO.

Upon receipt at the SPO terminals, the messages will be printed and enveloped, processed through the normal mailstream, and delivered by postal employees to postal customers.

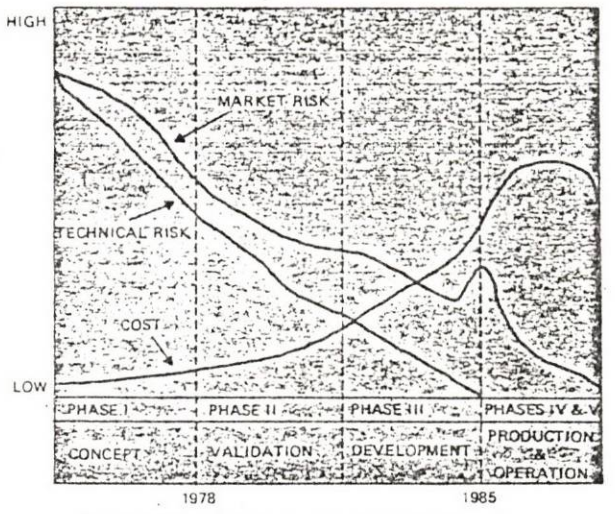


Fig. 1 EMSS Risk Factor Analysis (By Phase).

A PHASED APPROACH

Looking ahead, the USPS is taking a phased approach to Electronic Message Systems. This involves five phases, each with different cost, market, and technical risk factors (See Figure 1).

The concept phase, which is now being completed, includes the following:

- System and service definition and evaluation
- System and service description and architecture
- Service potential
- Resource Projection
- Social, economic, and regulatory impact evaluation.

This phase has required a high technical and market risk, but relatively little cost (about $10 million).

The second phase — validation — is now proceeding. It includes:

- Development of a detailed system design
- Testing and evaluation of the electronic message system
- Technological concept, design, and specification tests
- Market research
- Development of a multi-city network as a test bed.

The validation phase, scheduled to run for three to four years, will cost $55-60 million.

The third phase — development — should begin in the early 80's, with the groundwork for hardware and software design having been laid in the earlier stages. This stage, of course, will be very costly, but the technical and market risks should be substantially decreased, if the USPS has done its homework previously. Indeed, initiation of the development stage represents a proverbial point of no return.

36 ▶

The fourth and fifth phases — production and operation — are scheduled to begin by 1985.

How It Will Work

The proposed system would accept electronic input — generated by facsimile, computer, mag tape, terminals, etc. — in adding to standard hardcopy letters. Once the letters or messages were input into the system, they would be "translated" into bit streams and processed by a computer — perhaps stored — before being electronically delivered, either to a local Post Office (where they would be output in hard copy for physical delivery) or to the recipient's own network connected terminal. The Post Office "nodes" would be connected via satellite and wide-band terrestrial links, and message transmission may include pocket switching.

Utilizing the store-and-forward capabilities in the computer system, several varieties of priority message service will be available (from one hour to one day), in addition to such services as multiple address delivery of specified messages. The computer will also handle routing, switching, assembly, error checking, etc.

The Technology

Technological assessment is being handled by Arthur D. Little, and system support by the Naval Electronic Labs, the Institute of Telecommunications Science, and the National Bureau of Standards. RCA is coordinating these efforts, and providing product definition and market analysis. The USPS will, however, select prime contractors (based on recommendations), and will manage and control the system.

For hardware, the USPS is looking for equipment that will eventually be able to handle ten pages/second. Pitney-Bowes has developed a paper handling device that will accomplish this, and a prototype is currently operating at its Stamford, Connecticut headquarters. Fairchild is developing a scanner that will manage ten page/second throughput, while Versatec's electrostatic printer will be interfaced with the Pitney-Bowes equipment to handle the ten page/second output. The USPS will realize nine patents out of this state-of-the-art equipment.

To be sure, this equipment has not yet been integrated and fully tested. Yet, scaled-down prototypes will be available to begin the validation and testing phase next year (using four pages/second equipment). The central processor, system software, and some type of "electronic mailbox" for storage have yet to be contracted — but they are already available in various forms, and require little specialized development time in comparison to the printer/scanner/paper-handler.

The equipment for the laboratory demonstration for the validation phase will be installed in the USPS research lab in Rockville, Maryland, but is expected to handle only four pages/second for now. Three different types of distribution will be simulated: between addressees served by a single Post Office, between addressees using different Post Offices in the same city, and between addressees using different Post Offices in different cities. Leased loopback circuits, extending from Rockville to Los Angeles and New York, and back, will simulate the intercity distribution scheme. Terminals to be used in the demo will provide facsimile, mag tape, and printer I/O.

Looking Ahead

The "electronic mailbox" unit will be installed in Post Office lobbies, and possibly in shopping centers for hardcopy input. It will also collect the postage, and convert the hardcopy via OCR/fax to a binary format (e.g., magnetic tape). This input will be collected (perhaps "polled" by the USPS central computer) periodically. Only Post Office lobby units will be used in the validation experimental phase, which will use six to ten Post Offices.

Ultimately, the USPS hopes to *supply* EMSS customers with a variety of system-compatible terminals for input. This, of course, could raise some regulatory flak from terminal suppliers, although the USPS intends to make the network accessible (via code converters) to all popular terminal types. Computer I/O transmission may also be handled.

Developmental Goals

As a result of the RCA study, the USPS has outlined *nine* key system/service design parameters that have been used to develop and to evaluate potential EMSS delivery "candidates":

- Message Volume
- Input Conversion Technique
- I/O Media and Conversion Devices
- Number of EMSS Stations and Nodal Centers (number of public terminals)
- Depth of EMSS Sort
- Telecommunications Network Choices (terrestrial, satellite)
- Service and Time (priority — overnight, 1st and 3rd class)
- Privacy and Security
- Error Rate, Maintainability, and Overall Performance.

Based on these parameters, three "candidate" service delivery systems have been developed for further study. These are summarized in Table 1. Currently, candidate "C" appears to be the most viable. It is interesting to note that the USPS anticipates most of its volume will come from *hardcopy* input, as opposed to electronic. (This is a valid assumption. Why should terminal users go through the USPS when they can send messages directly?) No EFTS has been planned as yet.

Analysis

The USPS has mapped out an elaborate system for EMSS that looks fantastic on paper; there is no doubt that the Postal Service has to step into the electronic age

TABLE 1

SUMMARY OF USPS EMSS CANDIDATE PARAMETERS

CANDIDATE	ANNUAL VOLUME—OUTPUT IN BILLIONS	NUMBER OF FULL I/O PROCESSING STATIONS	NUMBER OF PUBLIC INPUT TERMINALS	STAFF	COST PER 1,000 MESSAGES	TOTAL INVESTMENT (IN MILLIONS)[1]	OPERATING COSTS (IN MILLIONS)[2]
A	19.8	150	1,986	9,177	$26	$1,375	$381
B	22	360	6,225	9,506	29	1,913	442
C	24.1	87	7,114	3,942	18	1,590	285

[1] Hardware, facility, engineering

[2] Manpower, maintenance, consumables, energy, transmission channels

Source: USPS

38

for message delivery, but the problems it faces are far more complex than simply developing a fancy electronic message system with state-of-the-art technology. First of all, it is rather naive to assume that the USPS will be able to make substantial cuts in its labor force. In addition, there are all sorts of regulatory and social problems to be worked out, as well as utilization dimensions.

On the regulatory side, the USPS and the Postal Commission will definitely have a *tete-a-tete* with the FCC regarding the extent to which the USPS can offer and/or monopolize electronic message services. Indeed, the Postal Service Act of 1977 — HR 7700 — in its unamended form, could have given the USPS a mandate to expand its services throughout the communications industry.

Moreover, the bill called for the USPS to "develop systems for electronic transfer of messages," by spending "at least 2% of yearly USPS revenues by 1980." This could have amounted to $300 million by 1980! Under pressure from the Computer and Communications Industry Association (CCIA) and the Van Deerlin Communications Subcommittee, some of these provisions have been amended, others deleted. Nonetheless, the USPS's proposed EMSS system creates some complex policy issues concerning competition and regulation.

When does an electronic message become a letter?

According to the Private Express Statutes, an electronic message sent to a person or specific address is not a letter . . . until it becomes a *tangible object*. Therefore, when or if the message is output as a hardcopy, that could, depending on the message and *means* of delivery, be construed as a letter — and become subject to Private Express Statutes and regulations.

What does this mean?

Could the USPS gain a monopoly over electronic mail? Should it have a monopoly over such service? Should it be allowed to develop and market terminal and switching equipment? These are some of the tough questions that must be answered over the next couple of years. There is, however, no question that the USPS's proposed EMSS will definitely impact carrier revenues (particularly AT&T's and the IRC's), since the USPS will be buying at a bulk discount. These carriers now handle virtually all of the electronic message traffic.

Besides, the carriers (specialized and value-added) may offer *better* services than the USPS. AT&T's Bell Data Network/Advanced Communication Service (BDN/ACS) promises to offer terminal users an excellent channel for electronic messages. The network will include intelligent switching, code conversion, error correction, and possibly even encryption — the same sorts of things that the USPS is looking to provide. Will the USPS go into the carrier business against AT&T? What will the government have to say? Who's going to pick up the pieces. . .of mail?

Also, terminal vendors will be scrambling to get on the USPS "preferred" list. Those who aren't will put up a vociferous volley that is sure to end up in the courts. Moreover, from a growth and development perspective, the electronics industry has done tremendously well for itself with a minimum of government interruption. It would seem likely, based on past performance, that a government policy of direct participation in electronic mail and messaging services would bode ill, at best, for successful propagation and efficient utilization. To be sure, why should business terminal users with access to communications, *pay* the USPS to "massage" their messages? What may happen is that the government will "tax" private electronic mail networks and message services to subsidize and encourage use of the USPS service.

Finally, we must consider whether the USPS will be able to *sell* its ambitious program to consumers. Frankly, given the recent past of the USPS, we doubt that business will buy it. . .unless it is forced to do so through government intervention/regulation. The home market is too small to worry about. Besides, by the time the USPS gets its act together, inexpensive hybrid fax/OCR/terminal devices will be available for home use — and communicate either through cable TV, or via phone lines.

Facsimile Electronic Mail

Xerox, the copier king, will likely introduce two new facsimile models before the end of 1978 — a two minute desktop unit that will lease for about $90/month, and a subminute analog unit that will lease for $350/month.

Xerox, which still holds over 50% of the installed facsimiles, has not introduced a new unit for a few years now. In the meantime, the market *and* the competition have changed significantly . . . to the point where Xerox's entire line of fax machines is no longer price competitive.

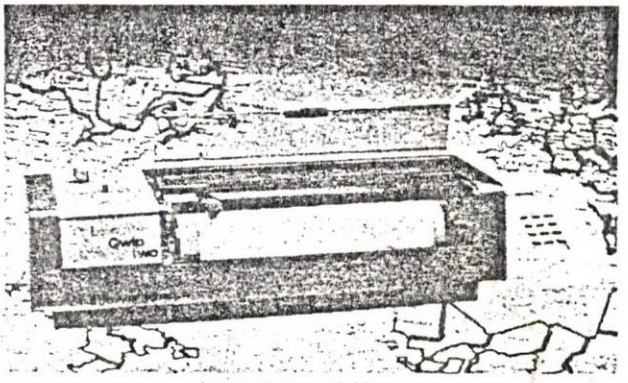
At the high-end of the market, Xerox presently offers only the TC 200 (a 2-minute unit) to compete with sub-minute machines from Rapicom, Panafax, 3M, Graphic Sciences, *et al.* And Xerox's 2-minute unit is too high-priced for today's 2-minute market place.

THE COMPETITION

Qwip

Qwip has introduced its 2- and 3-minute unit that leases for $65/month. Resolution is fair — 78x96 lines/inch at 2 minutes, 96x96 at 3 minutes. The Qwip 2 has an interrupt feature, but it is *not* compatible with the earlier Qwip models (1000 and 1200). It does, however, conform to CCITT Class II standards.

Qwip will also be introducing an automatic receive unit by the first quarter of 1979. Later in 1979, or in early 1980, it will offer a sub-minute entry and a store-and-forward broadcaster.



Qwip 2 from Qwip.

There is little doubt that Qwip is now interested in selling the large user and large national accounts, as opposed to simply peddling to the "33 million small businesses with telephones." It is also apparent that Qwip still has every intention of increasing its 12% share of the facsimile market . . . even if it has to spend 0.001% of Exxon's money to do it. Presently, Qwip has a negative cash flow approximating $10 million per year.*

* According to Howard Anderson, president, The Yankee Group, who tracks the company.

40 ▶

Rapicom

Alias Rapifax and Dacom, both owned by Ricoh; they have now combined into Rapicom. Although the Rapicom 100 is five years old, it is a good dependable machine, and it hasn't faced any serious competition in the high-speed fax market until recently . . . and now there's 3M, Panafax, and Graphic Sciences on the doorstep, with Xerox, Qwip, and possibly IBM looming at the back-door.

Last year, Rapicom introduced the System 50 — a fax controller that can automatically dial up to 50 locations. At this year's ICA show in May, the firm unveiled its Rapicom 650 — a facsimile message switching system that can be interfaced into existing message switching networks . . . and enable *true* store-and-forward facsimile switching. The Rapicom 650 consists of a Rapicom 150 (old Dacom unit with a few changes) facsimile, a 650 fax controller, an inexpensive keyboard/printer, and a great deal of software. The 650 will communicate with message switches in bisynchronous, HDLC, or SDLC protocols. For large users, it should be well worth the $15,000-$20,000 purchase price.

Rapicom is also developing a four-second fax machine with a new MPU and faster coding algorithms that will operate at 56 kbps. It will be aimed at SBS and private microwave users.

Graphic Sciences

This Burroughs subsidiary has always had a large facsimile product line that has done moderately well in the market place, but the firm had fallen off somewhat in recent years. Earlier this year, however, GSI introduced a complementary line of sub-2-minute facsimile units that fit into its DEXNET facsimile network.

The DEX 1100 is a 2,3,4, and 6-minute "desktop" unit that comes in three models, all of which are transceivers that can communicate with other DEX models. The 1100 leases for $62/month. Some of the units comply with CCITT standards.

The DEX 5100 is a 30-second/page unit with a number of features, including full duplex, auto dial, and automatic receive/send. It can, however, be slowed down to communicate with other DEX machines, such as the 4100 2-minute "mailroom unit" and the 1100's.

Thus, users can establish true facsimile networks around the 5100, using the cheaper 1100's in remote locations, as well as any other DEX family members. This represents the first such facsimile network approach by a manufacturer.

GSI has all but abandoned it's slower speed DEX 1 units, even though there are a number still in the field. The firm is betting that the new "slow speed" facsimile standard is going to be 2 minutes. Qwip agrees, and seemingly, so does Xerox. We too concur.

Panafax

This partnership between Matsushita (51%) and Visual Sciences (49%) has already done quite well this year marketing its MV1200 — a 2,3,4, and 6 minute machine that matches CCITT standards, and rents for $70/month plus meter charge (probably $85/month).

Panafax's really *hot* unit — the UF 20, a 20-second machine — has not yet been delivered, although there should be shipments within 90 days. Another subminute unit, the UF 320, which leases for $275/month plus meter charge, should also be available soon.

Portending the future at the ICA show, Panafax also demonstrated the Model 2200 (prototype) — a 20-second digital machine with a built-in broadcaster that can send documents to 25 selected locations. The 2200 also has other "goodies", but its cost — $500/month — may severely limit its market appeal.

Also on hand was Panafax's desktop subminute unit — Model 7000 — which can handle four different speeds,

COMPRESSION AND OCR-FAX

Because facsimile is essentially a "dot reader" — the scanner "sees" the page as a matrix that can vary from 67 x 76 "lines" or "dots" per inch to a 200 x 200 dots/inch (nearly letter quality) or even higher, if more resolution is desired — that transmits "black" or "white" as it scans the page, it is obvious why fax transmission takes so long. That is, with reasonable resolution — *e.g.*, 100 x 100 — a standard (8½" x 11") fax-read page will have nearly one million bits to transmit, plus formatting, code, and digital-analog-digital bits; and since most fax machines operate over 2400 bps lines, a page requires around six minutes.

Thus, it is not the speed of the *scanner* that is slowing down the fax, but the transmission speed. This, of course, will be ameliorated with the availability of reasonably priced wide bandwidth transmission media. But in the meantime, fax manufacturers have mitigated this problem by developing techniques to "compress" the information ("bits") that is scanned *before* it is transmitted. That is, most of the information on a page is redundant — primarily white space — so compression algorithms transmit only the first and the last "white" or "black" bit in any sequence of white or black space. These techniques make transmission 10-15 times more efficient, depending on the exact algorithm utilized. Virtually all high-speed fax machines use some type of "compression coding". Manufacturers are constantly on the lookout for better coding algorithms that will further cut the speed of their facsimiles.

One solution that has R&D people drooling is the prospect of an optical character reader (OCR) — facsimile hybrid. Since an OCR "recognizes" scanned characters, and transmits them in 8-bit "bytes", a page requires only 5,000-6,000 "bits" for transmission (about 2 seconds on a 4,800 bps line). Moreover, the *output* is much "cleaner" using OCR. Since most fax transmissions involve *text*, OCR is an obvious solution.

The problem is that current technology is limited. Reasonably priced ($15,000) machines can read only a handful of specialized fonts. There are OCR machines, such as the one Fairchild is developing for the USPS and an REI device that can read almost *any* font — but they cost over a few million dollars. This situation is rapidly improving, since "recognizing" more fonts is a function of using larger and faster, but cheaper memories and processors in the OCR device — and these developments are occurring even as we speak.

Stewart Warner's $20,000 OCR-fax device that was announced last year was heralded as a fantastic breakthrough, but a real-world model has yet to see the light of day; it may likely never be available from that company. Industry experts to whom we have spoken indicate that such a machine (a true OCR-fax hybrid that can handle any type of font) at that price is not feasible today. There is, however, no doubt that some type of OCR-fax device will be *the* source document transmission terminal of the future. We think that by the mid-1980's, such units should be commercially available, and by the 1990's, OCR-fax "type" terminals will be *fully integrated* with CRT/keyboards and non-impact printers . . . as part of an individual's completely interactive, intelligent work station. For the present, however, we must work with the "fax" that is (are) available, and better data compression is the name of the game.

42▶

and would lease for $175/month + meter (figure on $250-275/month). This impressive little box uses plain paper and inkjet printing, as well as a new compression algorithm. It is, however, not yet available in the US.

We have heard from a reliable source, that Matushita (renowned for its conservatism) may be having some second thoughts about marketing in the US . . . spending another $40+ million in development before seeing any profits. Also, the yen's recent revaluation against the dollar (195 yen/dollar) puts a squeeze on Panafax prices, which are based on 230 yen/dollar.

Compression Labs

Compression Labs has developed a highly efficient, unique coding algorithm; its post-recognition processor "recognizes" (by referring to its memory) repetitive *patterns* as they reappear on a given page. The device then transmits only the patterns (which can, like characters, be defined by a handful of "bits"), thus substantially reducing communication time — and line charges. The Compression Labs device, which is currently being marketed as an add-on "black box" called FAX-COMP, is in a sense, a "learning" Optical Character/Pattern Recognition machine.

The FAX-COMP can reduce a six-minute/page transmission to under 30 seconds. The basic unit includes a telephone handset, a 4-prong telephone system plug, a 12-key Touchtone numeric pad, 3 pages of storages, 2,400 bps modem, automatic dialing, *et al.* Options include a 9,600 bps modem, a floppy disk with *10* pages of storage, store-out forward messaging, remote site polling, and LED display.

It costs $4,500, and can be interfaced to Xerox 410 and 400, or Qwip 1200 facsimiles; optional interfaces are also available for GSI, 3M, and Xerox 200 machines. Of course, a FAX-COMP is needed at both ends of the transmission. The device will certainly extend the life of these four and six minute machines (*un*obsolete them).

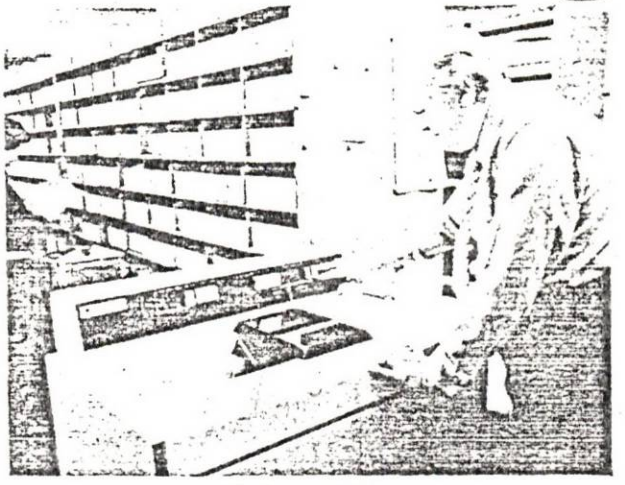
Compression Labs is presently developing its own facsimile machine.

3M

The Express 9600 is alive and well, and actually *working* on 9600 bps, untreated lines, although it often has to drop down to 7200 bps to get clear. This 20-second/page digital unit has three major drawbacks: strictly mediocre resolution (100 x 100) that can't be improved because the scanner can't slow down to "see" more; incompatibility with other 3M machines (again, can't slow down to meet them); no "go to the next address" provision if for some reason there's a transmission shutdown in the automatic send/receive mode.

Telautograph

An old hand at facsimile and "electronic writing instruments" which has been somewhat reticent in recent years, Telautograph will soon announce a one minute analog machine (meeting CCITT standards) with a *true* thermal printer . . . for under $150. Hitachi is known to



3M Express 9600.

have developed a similar machine. In fact, there could be some licensing or distribution agreements between Hitachi and Telautograph. Question: Has Telautograph solved the heat building problem in the thermal head? If so, this unit should "heat up" the fax competition, assuming Telautograph can find some way to market and service the machine. It'll be a little different than selling "electronic pens".

IBM

With product life cycles aimed at five to ten years, and an enormous R&D staff, IBM has to do something in the "in-between" time. The Research Triangle down in North Carolina has developed a prototype *three-second fax machine* that it uses to communicate (via wideband links) to New York. This speed is no great feat at wide bandwidths . . . but IBM's toying with fax is definitely significant. Indeed, "toying" may be the wrong word. IBM has reportedly retooled some of the Copier 1's it has laying around with fibre optical tubes . . . and turned them into *Non-Impact Printers*! These, in fact, may be the "receivers" for the 3-second fax machines, among other possibilities.

We think that IBM is still trying to ascertain the potential size of the fax market before entering. The Armonk giant has a propensity for only choosing BIG markets that it can dominate. We have heard, however, that an IBM fax machine (or two) is being "BETA" tested out there somewhere . . . maybe Canada.

By the way, what might IBM want with 30,000 of Paradyne's 9600 and 4800 bps modems . . . over the next three years?

The Future

We know of a few manufacturers. NEC and OKI among them, which are developing fax/CRT prototype systems that will enable interterminal transmission. After all, a roster scanner does "play" with dots.

Computer-Based Message Systems

THE MARKET

The market for Computer-Based Message Systems (CBMS) today is tiny, but it will shortly become the fastest growing area of electronic mail for two specific reasons. For one, CBMS is more *user-oriented* than traditional electronic mail services and is far more *productive*. For another, the announcement of AT&T's Bell Data Network will enable the widespread use of Computer-Based Message Systems with a minimum of problems.

Let us try to defend these two statements.

What Is A Computer-Based Message System and Why Is It a Special Form Of Electronic Mail?

A true Computer-Based Message System is a special form of electronic mail because it allows the user to access incoming messages at his/her convenience, to dispose of them electronically, and to file or pass them along as he sees fit. It leaves a perfect audit trail, and eliminates or at least reduces the need for paper files.

44 ▶

How does such a system increase efficiency? And how is this efficiency measured? What are the impediments to a Computer Based Message Systems environment?

Advantages. It allows people to handle some 30 pieces of routine correspondence in as little as an hour. This compares with the ability to make some 5-6 telephone calls in that time, hold perhaps one meeting face to face, and write perhaps three letters. It also allows managers to use NON-SIMULTANEOUS TIME to solve non-simultaneous problems instead of real time.

It puts problems into perspective. Problems can be handled relative to their importance, and during a specific period set aside for problem solving.

A CBMS allows for interruption-free periods. Psychologists claim that we work best when we have some period (45 minutes - one hour) of uninterrupted time where we can concentrate on the task at hand. This is an anomaly in the modern office. The telephone is a constant interrupter. If an executive is "out" then those pink call-back slips pile up like so many pancakes. When calls are finally returned they often go to empty offices or to offices where the other party is in meetings or is unavailable. In short, "telephone tag."

Users on a CBMS find that the number of incoming phone calls significantly decrease. They find that almost every day is like the working Saturday when "I got an unbelievable amount of work done."

Disadvantages. To begin with, a CBMS really works best when the originator keys in inquiries and responses. Today's executives are not excited by this prospect. To say the least, neither are managers, especially women managers. Most executives really are reluctant to have "keyboards" or "computer terminals" in their offices. They regard the terminal as degrading, and resist the intrusion of such a bulky, ugly device into their plush quarters. There is however, a marked difference between executive attitudes relative to their age and to their industry-orientation. Younger executives are more likely to use such a terminal than older executives. Firms which are into automation and whose executives and managers are accustomed to keyboards are more willing to implement such a system. Examples: Digital Equipment Corporation, NCR, IBM, Bell Northern Research.

If we view management as a pyramid shaped triangle, CBMS may work best in the middle third, roughly comparable to middle management. From the software/hardware point of view, this is even more attractive as a market than "top management", if only because the potential is greater for terminal placement.

TYPICAL SCENARIO FOR IMPLEMENTATION

Many companies use the following scenario in implementing an electronic mail system:

- Establish a task force to investigate the needs and uses of an electronic mailbox system.
- The task force makes a recommendation for a trial test with off-the-shelf software, which is contracted for by an outside vendor (Computer Corporation of America, Compuserve, Scientific Time Sharing, Tymnet, etc.).
- Management agrees and trial system goes up with a small, predefined group of less than 25 subscribers.
- Trial period concludes. Additional modifications and additional subscribers are added to subsequent trials. In-house task force monitors use very carefully.
- Trial period gives way to decision. Should the company proceed full steam ahead? If so, who will pay for its use?
- Free users face choice. If they loved the system when it was free, will they love it when it costs? Users must decide and put it into the budget.

At this point, most experiments are getting knocked down.

Most of the other costs — internal mail, telephone, etc. — are usually not costed back to the using department. They are part of corporate overhead and are billed according to some formula (number of employees, number of square feet, etc.). But CBMS's are very identifiable; costs come in each and every month, and users get no adjustment or relief from the internal mail costs. They must pay the full cost, if they want it, of an electronic mail system.

COST/MESSAGE FOR CBMS

What are those costs? For an off-the-shelf system, excluding terminals, the costs run some $0.75 - $1.00/message. Assuming $0.75/message, the average user will transmit some 15-20 messages per day ($11.25 - $15.00) or $2500+/year/employee. Most firms that we survey use a cost of $1500/year/subscriber as a minimum.

Further assuming a middle management level at the $30,000 level, the CBMS represents a cost of 5-8% of individual payroll. Management asks, "Do we really get an increase in efficiency of 5-8% better?" The users answer: "Yes, we are easily that much more efficient, but you shouldn't even look at this cost because it replaces other costs — secretarial time, telephone, postage, filing. Even more important, it gives us better information faster. It helps you and us to run the company better." Management's answer is succinct: "Prove it."

This is difficult. Even if they wish to pay for the system out of their budget, this option is not always open to users. In effect, they're asking management to make a leap of faith. Management tells departments to limit their capital appropriations for the coming year, and the Computer Based Message System is ripe to be cut.

THE SPAN OF CONTROL ARGUMENT

The best argument that the user has is not savings in secretarial time, paper, filing cost, telephone expenses, or even his own increased productivity, but in *increased span of control*. This Citibank hypothesis is that a company needs *fewer* executives to perform supervisory tasks. That is, a reduced number of people can, with the aid of electronics, perform the necessary work, thereby increasing the *span of control*. In Citibank, the present span of control is seven subordinates for each supervisor. The company today has 49,000 employees. With normal growth, that figure will reach 55,000 in five or six years. If the span of control could be increased to 8:1 or 9:1, there would be no need for additional executives, and the savings would approach some $150,000,000, according to Citibank.

Of course, these numbers border on the absurd. But if Citibank could cut its expected middle management hires by 1,000, this alone would result in a savings of $25,000,000. The cost of such savings: approximately $10,000,000. (Having 4000 executives on an electronic mail system at $2000 each plus $2 million software and supervision cost = $10 million). Although this is a blue sky analysis, a savings of potentially some $15,000,000 is enough for top management to investigate the least *possibility* that such a scheme might have merit.

The real question: *Is it possible to reduce the number of managers, increase the span of control, and continue to run a well managed company?*

In short, no one knows.

In the past, executives wrote all their memos themselves in longhand. This was replaced by the secretary and the typewriter. As secretaries, dictation units, and electric typewriters became more pervasive, a curious thing happened. The amount of written material increased geometrically. And with the advent of reliable and cheap photocopying, the reproduction problem evaporated — and turned into a reading problem, a filing problem, and a response problem. Do executives manage better today because of office automation? Probably not. Do they manage better today because of computers? In some cases, definitely yes: in others, maybe.

At present, Citibank has backed off from its lofty and widely publicized goals. For one thing, earnings were down last year and Citibank management is hell-bent to regain its growth lead over Bank of America. For another, Citibank's management feels that perhaps it should curtail some of the effort and spending which was going into this project.

The idea may be sound even if the most visible experiments and experimentors aren't. Several officials in The Office of The White House are evaluating CBMS now, although not for span of control reasons.

Assuming that the hypothesis is true: that with advanced automation techniques in The Office (primarily a CBMS and some teleconferencing, etc.), it would be possible to run a tight organization with less people, and to do it well; *how could one test the hypothesis?*

The first step: to begin increasing the workload and simultaneously bring in new office automation solutions. If an executive or a manager isn't pushed to find a new solution, it is unlikely that he will.

The most immediate reaction by most managers to an increased workload will be to step up their traditional modes of behavior. They will try to handle more with the same methods previously used to handle less. They will attempt to begin earlier, stay later, and accomplish during the day. Using this approach, they will be only moderately successful. Going from managing seven executives to nine executives is an increase of almost 30%. And we are not talking about just *one* manager. At each level, managers will have to supervise more subordinates; this will be true throughout the organization.

Within six months, it will occur to our fictitious executive that two things are happening. For one, he is going to be judged on how well *he* can cope with this increased workload and how well the people he manages can cope with this increased workload. In short, even if he is successful, his promotion may depend on how well the people he manages cope.

The second thing: the manager needs a new style and method of management if all this is going to work. At this point, he will begin to use some of the management tools which he has avoided to this time like the plague. He will have given up on parts of the old system and begun to rely on the new technology. He will realize that the morning "get together" in his office is an extravagant vestige of past practices, and that his subordinates cannot afford this kind of waste unless it is a special occasion. He will begin to utilize CBMS to insure that everyone is informed of his intentions without having to hold a general meeting. He will also begin to insist that if these new tools are good enough for him, they are good enough for his people.

Before we get carried away with this blue sky thinking, let us say that without some motivation to make this happen, it won't. Managers are like the rest of us; they get their enjoyment more from interaction than flying desks. The theory of goal congruence says that when a manager is faced with a decision of doing what is good for his company or with what is good for himself, he will invariably choose the solution which is best for him. Top company management today spends some 80% of their time in communications. Middle management spends some 60%. Management is judged on many things; the ability to use office automation equipment to become more productive is not one of them. Managers get ahead in their companies by good politicing and by keeping their programs on course. In effect, managers don't manage at all — they control things. The short-term prospects are not good.

But the long-term prospects are better. All in all, companies are in the same business — making money. If one company in an industry finds an innovation that allows it to run better and leaner, it should be able to grab more of a market share, run higher profits, hire more capable executives and far outdistance its competition. That is, in theory.

In actuality, a firm needs (and an individual needs) more of a push. The push may come from bold and successful action of a competitor or from the realization that the use of these tools may further personal advancement.

This is where we stand today. CBMS's are expensive, but powerful. There is no question that they provide the ultimate in immediate message transfer.

The question is, "*Are they worth the cost?*"

CURRENT CBMS USERS

A number of firms think that perhaps the answer to costing is yes, and are willing to investigate this concept on an experimental, or limited basis: Citibank, Digital Equipment Corporation, Shell Canada, General Motors, IBM, Bank of America, Hewlett Packard, The Arpanet, The Department of Energy, The Department of the Army, Honeywell, General Electric, and some 22 more firms that we keep track of at The Yankee Group.

The market today has less than 5000 terminals dedicated to electronic mail, Computer Based Message Systems variety.

We expect this market to grow by at least 100% during the next three years, as the original experiments take root and expand and other companies begin their experiments.

The table below reflects not only public access systems but also private intracompany networks. A company with only a reasonable number of subscribers on an electronic mail system would be foolish to develop its own; what is commercially available should suit its needs. However, once the number of subscribers goes beyond 100, the costs of going outside are beyond reasonable.

Computer-Based Message Systems

1978-1983

Year	Number of Computer-Based Message Terminals	Number of Messages
1978	5,000	18,000,000
1979	10,000	40,000,000
1980	19,000	79,000,000
1981	36,000	150,000,000
1982	65,000	275,000,000

(The Yankee Group)

The costs break down as follows:

From an outside vendor
Charge per message: $.70-$.75
His costs:
Computer cost/message: $.20
Communications cost/message: $.10
Overhead, file update: $.05

Gross Profit per message $.35-$.40

These costs are essentially the same for a large user who wants to go on his own. The only problem is the software package, that's expensive.

But a number of firms will lease their software for the payment of a one-time fee plus yearly maintenance. CCA's are $40,000. Stiff but not ridiculous. Suppose a company has 200 users sending 3600 messages per year at $.70/message = $504,000. If it can essentially cut that cost in half with a payment of $40,000, this is a cost-effective decision. Also a larger firm can run its computer-based message system from its existing computer nodes, capacity permitting.

As an industry, the Computer-Based Message Services Market is in its infancy. Compuserve has set up a new subsidiary, Plexus Corporation, to begin marketing its version (Infoplex). Compuserve already has over 700 subscribers on its system, including General Motors (400) and NCR (200). Computer Corporation of America has about 200 subscribers and four accounts. Typical account: Shell Canada, OnTyme (Tymnet) about twice that; same for Umex. Scientific Time Sharing is dickering with the FCC about its status. Telenet has its Telemail system which really isn't being commercially marketed yet, but will be shortly.

POTENTIAL SUPPLIERS

Who is interested in this market? Virtually everybody. Tymnet got into this market because its users on TYMSHARE (time sharing) were inputting electronic messages at one location and outputting at another. Two years ago it made the decision to become a common carrier because it realized that it was really in *that* business, and didn't want to run afoul of the FCC. Compuserve spun off this company, thereby giving it the opportunity to become a common carrier at a later date.

One company which has only toyed with the idea of becoming a common carrier is General Electric. General Electric Timesharing is the largest in the country. GE has access ports in over 500 cities (Tymnet in about 190). General Electric is well aware that some of its users are using timesharing now as a cheap way to send electronic mailbox messages (albeit illegally). It recognizes this opportunity as a potential market opportunity, but it also realizes that its timesharing networking is really quite dumb by current standards, and requires some technical upgrading. General Electric's situation is this: If it entered the "electronic mailbox" market, it would have to become a full fledged Value-Added Carrier. If GE did become a Value-Added Carrier, who would its customers be?

Obviously the first customer would be GE Timesharing. In effect, it would be at "breakeven" from Day One. Secondly, GE would be able to convert a number of its existing timesharing customers to its Value-Added Services (such as electronic mail) from the very beginning, because these customers are using the system for this now. Thirdly, because of its access in so many cities, GE would have immediate advantages for users trying to find the right Value-Added Net.

Who else is interested in this market? Well, AT&T for another. AT&T's major problem is finding highly profitable services to make up for all the lower rates it must charge for regulatory/political reasons.

If one looks at the AT&T income statement, one sees that of the $36 Billion in income, some $11 Billion is revenue to AT&T Long Lines. This is the most profitable part of AT&T and the one, if the truth be known, that supports the entire shooting match. Of that $11 Billion, $4 Billion is Business Interstate Message Toll Service, $1.4 Billion is WATS (a business service), and $1.6 Billion is Private Line (business service also). The remaining $4 Billion is residence message toll service.

In short, business is the major customer of AT&T Long Lines. And it is directly at the heart of this business interstate market that AT&T's major competition is aiming — MCI (voiced), Southern Pacific Communications (voiced, low-speed data, facsimile), and even Satellite Business Systems (voice, data, image).

Not to be forgotten is ITT. ITT has received approval for the COMPAK service (facsimile and keyboard terminals). Each of these services *bypasses* AT&T Long Lines. In short, they neatly excise the most lucrative part of AT&T's revenues.

Furthermore, these competitive services ultimately weaken local telephone companies, because a disproportionate amount of interstate call revenues goes back to the individual telephone companies through something called *separations*. This is not an insignificant number; some independent telephone companies (there are some 1600 of them nationwide, the largest being the various GTE companies, United Tel, Rochester Telephone, etc.), receive as much as 60% of their *total revenue* from separation. The Bell Operating Telephone Companies (New Jersey Bell, Ohio Bell, Bell of Pennsylvania, etc. — 23 in all), have similar revenue scales.

It is not an understatement to say that the local telephone companies and the Bell Operating Telephone Companies are *loss leaders* and that the real money is made in long-distance telephony — a fact that AT&T's competitors are well aware of. And one of the main *selling points* of not only the other carriers, but the new Value-Added Carriers, is *electronic mail services*. Besides Southern Pacific, ITT Domestic Transmission Services, Satellite Business Systems, Western Union, etc. offering such service, the Value-Added Carriers will, and are offering electronic mail services (Telenet, Tymnet, Graphnet).

These Value-Added Carriers have no loyalty to Bell. They can put their requirements up for bid and avoid the Bell System wherever they have a more competitive bid, using Bell only where there is no alternative.

Long Lines is under intense fire. As AT&T looks at the market for Computer Based Message Services, Value-Added Nets, etc., it sees a highly lucrative opportunity, one that is more profitable than the local telephone exchange business, and one which can produce a substantial premium over service cost.

Enter Advanced Communications System (ACS). One of the more obvious applications for this Value-Added Net is Electronic Mail because the service will include message routing and message rerouting and there will be buffering within the system.

We don't want to make ACS sound like it is an exclusive electronic mail service. It isn't. It's designed primar-

ily for mid-sized, and even large users. How will users use ACS? Basically they will be able, at a lower level, to perform some of the functions now done by their front ends, multiplexors, concentrators, and remote concentrators.

It will enable the user to do error checking, network control, speed and code conversion, message routing and rerouting. It will give the user a degree of control, expandability, flexibility, and transparency that is not now possible at a reasonable price.

The impact for Electronic Mail? Quite a bit. Bell is providing the framework for advanced Computer Communications Message Systems. Although we doubt that the Bell software packages will be as sophisticated as some of the software packages now available, these will evolve as users gain experience with the network and demand more functions and applications. The ability to offer these products is limited only by the decision to put software specialists on the project.

We believe that the inclusion of Bell in the value-added business in general, and the electronic mail business in particular is *not* a forgone conclusion. Each and every provider of electronic mail services, value-added services, and competing EDP firms will squeal like a stuck pig when Bell introduces the Advanced Communications System. But it is our opinion that Bell will be able to enter this market, although the regulatory flight may take three years.

In summary, the Bell System is interested in Electronic Mail because it provides AT&T with a highly profitable service with only a small increase in fixed costs. It allows AT&T access to a new growth market (necessary with the number of new phones only growing at 4%/year) and it provides it with a better chance to cash in on the profitable portion of data communications-high value data services.

Communicating Word Processors

DALE KUTNICK
The Yankee Group

Most manufacturers now offer some type of communications facility on their fastest machines (either as an *option* or as *part* of the system). Some Communicating Word Processors (CWP's) are highly advanced and flexible and can also communicate with *compatible mainframe computers, terminals*, Telexes, TWX's, photocompositors, and other CWP's. Some models offer unattended reception with stack feeders; a few sophisticated shared logic systems also offer store-and-forward "electronic mail". A number of display-type WP's offer CRT-to-CRT communications, so no *paper* is required. Others can communicate one document while the operator is working on another.

CWP's transmit data over phone or microwave lines, or via satellite at rates that vary (depending on the machine) from 60 baud to 9600 baud, although most transmissions occur at 1200, 2400, or 4800 baud. A full page *can* be transmitted in 6-8 seconds. Modems are required at both ends unless a digital line is being used . . . *not many* of those are around today. CWP's can also be used to enter material into a photocompositor, and in the near future, they will be able to send/receive information from centralized micrographics files.

CWP's will be an integral part of any electronic message system. They offer some distinct advantages over facsimile. These include:

- Superior speed. This is important because it saves on telephone line charges. CWP's handle "whole characters", which can be transmitted ten to twenty times more efficiently than facsimile "bits".
- Far superior output quality. Because fax *creates* letters or images via "dots", letters are not as clear and crisp as letter-quality typewriters used on CWP's.
- Hard copy may be unnecessary. When using CWP's material can be sent from tube to tube. Also, double-handling of generated text is unnecessary.
- Low cost. The communications option *alone* on a CWP is relatively small ($500 - $3,500).
- Compatibility. While CWP's can communicate with a number of other terminals, most of today's fax machines can *only* communicate with some other fax machines. It is difficult changing fax "dots" to terminal alphanumerics.

GROWTH OF CWP's

In the March issue of *The Technical Office* (our own in-house publication), we extensively discussed communicating word processors, their applications, utilization, and market and growth potential. We, of course, are updating our study and recomputing our numbers all of the time . . . to correct, or to reinforce our beliefs/estimates. After careful deliberation, our estimates of CWP's *in use* still hold, with only slight upward adjustments — some 13,200 in use at present, 17,300 in use by yearend. Actual shipments of CWP's, however, are doing even better than we expected, and we have revised our yearend 1979 forecast from 30,000 to 33,500.

Our other projections, through 1981, should also increase by about 10%, so that by 1981, we expect that there will be 138,000 CWP's, or communicating multifunctional terminals (CMT's), that are also used as WP's in use. We now estimate that this figure should nearly *double* by yearend 1982 . . . reaching 263,000 CWP's (or CMT's) in use. And by yearend 1983, we think there will be 379,000 CWP's (most of which, by this time, will be software-loaded CMT's) in use, as the WP/MT population moves across the 920,000 installed base mark.

The reasons for our revisions and great (some say optimistic) expectations? To begin with, there have been some significant developments — in CWP trends and introductions, in user acceptance and interest in CWP, and in the near-term potential for *intelligent* digital networks — over the past few months. The latter is a direct reference to AT&T's "Advanced Communications Service" that could become available by late 1979 or early 1980. . . . and go a long way toward curing what has heretofore been the stumbling block for inter-terminal communications — incompatibility.

BREAKING THE TOWER OF BABEL

The important CWP trend that has emerged over the last half year is that the CWP world appears concertedly

52 ▶

to be going to high-speed bisynchronous "batch" communications (2770, 2780, 3780) . . . and away from the slower speed 2741 asynchronous communications, which has seen extensive use on IBM's communicating mag cards (CMC/ST), on Redaction's Redactor I, on Xerox's 800, earlier Vydecs and Lexitrons, *et al.*

This mass migration to bisynch has *not* been a result of cooperative efforts among manufacturers — although there has been an increasing amount of dialogue when it's mutually beneficial (e.g., a user needs to convert code between two different machines) — but has come about "*de facto*" because it is a more efficient way of transmitting documents. Moreover, bisynch is one of the main languages of IBM's EDP world (bisynch can be easily massaged to work with SDLC), which, in itself is a good reason to adopt it as a standard. And besides, by using programmable or software loaded microprocessors as communications controllers, there's *no* reason why WP equipment can't support a number of interfaces, such as ASCII, EBCDIC, Telex/TWX *and* batch bisynch, as well as interactive 3270.

And this has begun to happen. Virtually all major WP manufacturers will support at *least* 2780/3780 bisynch. IBM's OS/6, Xerox's 850, Vydec, Wang, and Digital Equipment machines currently do, and Lexitron and Lanier (AES) intend to by yearend.

THE WORD IS MOVING — ELECTRONICALLY

What this means for the user is that his different machines will essentially *speak* the same language, with slightly different "code" dialects that can be programmed out.

For the vendors, compatible CWP's make the hardware more attractive because it is more versatile. For instance, since Wang announced its high-performance 5528 programmable communications processor earlier this year, more and more users have been taking the communications option. Over 30% of Wang's new orders go out with communications (and it's shipping 450 keystations — roughly 110 CPU's — per month). Xerox is including communications on 15% of its 850 shipments (575 WP's/month). Vydec is shipping 12% of its 350 units/month with the ability to speak. DEC customers are taking communications on almost 50% of their WS 78 orders (which are running at 90/month). And nearly 15% of IBM's OS/6's (790 keystations/month) are travelling with communications.

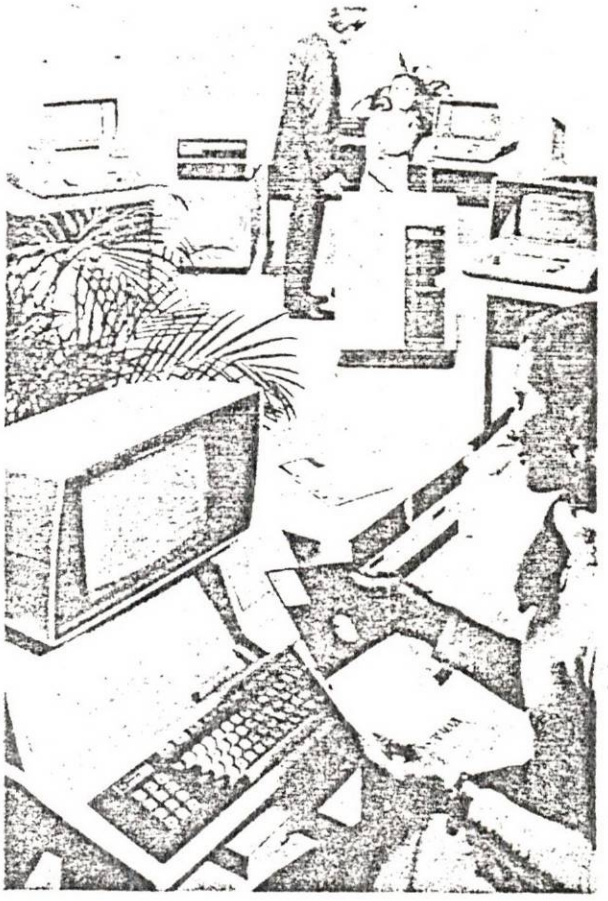
Moreover, Wang and Xerox CWP (and soon DEC) machines are fully duplexed — communications is a *background* operation, thereby enabling messages to be transmitted/received while WP operators continue text editing. This is *true* electronic mail, because it is *nonsimultaneous communication*. In the future, *all* communicating "terminals" will possess this capability. On a WP, it generally requires at least 24k-32k of memory.

Indeed, users are thus not only *accepting* communications, but are *demanding* at least the availability of an interface option . . . so that they can add "speaking power" at a later date. Moreover, some sophisticated users are developing their own applications — using CWP's as off-line editors for program documentation, COBOL source code creation and maintenance, etc. . . . before batching up to a 370.

Indeed, it is easier to perform editing on a "WP" terminal than on a 3270 screen . . . and besides the former is off-line! For these users, there is no such thing as a dedicated WP — only a protean terminal that wears a WP skin during the day, and an EDP mask at night.

One firm has Vydecs, Wang 30's and OS/6 talking to each other *and* to three 370/168's. Another uses a TI silent 700 as an I/O terminal for a Wang 30.

The CWP message is finally arriving. . . .



Texas Instruments Series 700 intelligent terminal.

THE NEXT STEP

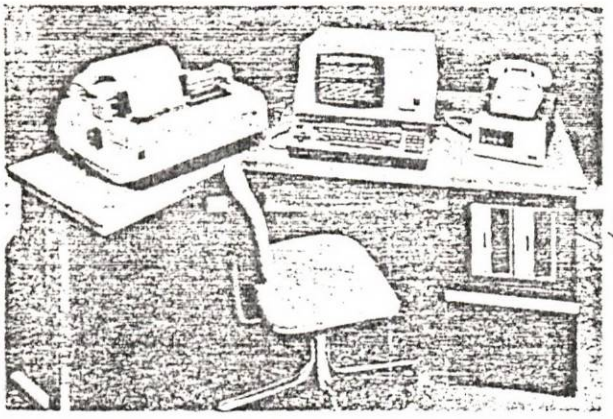
The next evolution for distributed networks is directly attributable to microprocessors — fully intelligent, multifunctional terminals with their own memory, and communications interfaces are being used for a "shared system" approach. In the office, these terminals are essentially stand-alone word processors with the ability to communicate. But for *intra-office* communications, these units can be interfaced *directly* (hardwired) to each other or to a large computer via a wideband medium that enables communications to take place transparently — the user *never* realizes that communications has taken place. A file on another machine in a different part of the building (that is part of the network, of course) appears as if it is adjacent. This represents *true* distributed processing, and is the forerunner to sophisticated electronic mail networks.

Datapoint's "Attached Resource Computer" (ARC), which was introduced late last year, is an excellent example of such a system. A number of discrete applications processors (such as word processors, file processors, small business computers, and even a 370 computer) can be "hung" on the high-speed databus (coaxial cable) to transparently share resources at *very* high speeds. Resource Interface Modules (RIM's) act as message switch-

ers in this mini-ARPANET-type network.

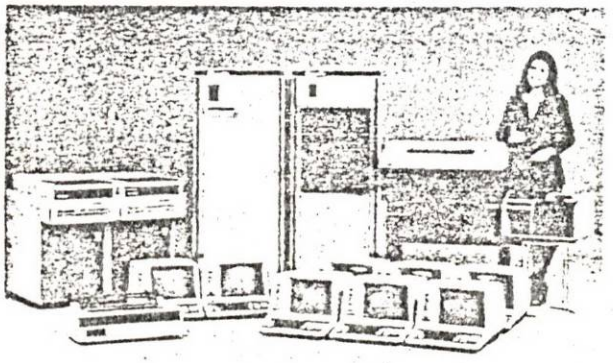
DEC recently introduced a WPS 11 interface that enables WS 78 or WT 78 stand-alone WP's to share resources (e.g., disk, files, peripherals) with a PDP11 computer, again at very high speeds. The PDP11 can, of course, continue to perform *other* functions (such as data processing, communications, etc.) with multi-leaving software.

Wordplex uses a similar linkup to interface its Wordplex 1 (via coaxial cable) stand-alone WP's to its Wordplex 7 shared logic (mini-based) system. This, however, is a *communications* interface, and is *not user* transparent. But, other stand-alone WP's with compatible communications facilities can *also* be linked to the Wordplex 7 mini (speed is 9600 bps).



Wordplex 1 from Dennison.

Lexitron will soon establish a "transparent" cable link between its Videotype 1000 WP's (which now have floppies), and parent company Raytheon's PTS 1200 (a powerful intelligent terminal with multileaving *and* 3270 emulation, as well as other protocols). The PTS 1200 will act as a cluster controller in this architecture, and handle DP functions, mass storage, and electronic mail as well, in addition to communications with mainframes.



Raytheon PTS 1200 Mark II.

DPT's new "Wordmachine" uses intelligent terminals linked to a "renovated," multileaved *360* with 500kb of main memory . . . to perform sophisticated word processing *and* electronic mail.

Wang is currently developing a "shared system" approach that will enable its stand-alone WP's to be linked to its System 30. This may be announced by the end of the year. Wang is also working on developing communications links that will tie its small business computers into a network with its WP line.

Vydec is developing a shared system approach using a new Z-80-based programmable standalone WP terminal . . . that will be transparently interfaced to a mini-computer/controller with 10 or 20 Mb's of Trident disk. The controller will also act as a message switcher of sorts, enabling the terminals to interact with a variety of other terminals and computers. Vydec's system, which will be announced by yearend, will give its users:

- More storage capacity (hard disk)
- More efficient, easier communications, particularly, with the Qyx, which is also Z-80-based
- The potential to develop an electronic mail network using Vydec and Qyx machines
- More flexibility (with a software-loaded terminal) and less obsolescence risk
- Better system expandability.

Xerox is "Beta Testing" a "shared system" in the federal government using its new "Alto terminal" (as in Palo Alto, where it was developed). It is still quite proprietary, so specific details are sketchy, but the system is known to include: a high resolution display, keyboard, a non-impact printer (xerographic, of course) that can handle graphics, and a 16-bit mini . . . in each "work station".

Work stations are linked on a coaxial cable data bus called the Ethernet, which enables 50-kbps communications. The work station/terminals — four of which are installed in the National Bureau of Standards, two in the White House, and one in Congress — can handle text processing/editing, electronic mail, and graphic functions.

Notwithstanding IBM's general lack of enthusiasm for distributed processing, we feel that it too will eventually take a "shared systems" approach to word processing. Indeed, a collection of OS/6's has excellent potential for transparent cable linkage to the Series 1 minicomputer but . . . .

GOING BOTH WAYS

The attractiveness of "shared" systems, aside from their multifunctional and shared file and peripheral capabilities, is that they incorporate the advantages of both standalone *and* shared logic WP's, while eliminating most of the liabilities. To wit, the WP terminals in true "shared" systems are essentially standalone (if the mini goes down, they can continue to operate — unlike shared logic systems "Christmas-tree-light," total-system-shutdown effect). However, the cost of the WP terminals in such systems can be substantially reduced (e.g., $7,000 for a CRT terminal with a floppy disk) because they can *share* printers and other peripherals, as well as the large disk memory of the mini. Moreover, the communications, storage, and file manipulation power inherent in the minicomputer (or intelligent terminal) node can also be shared by the WP terminals, thus opening all sorts of interesting network, electronic mail, and multifunctioning options.

Without question, these systems represent the beginnings of true distributed processing in the office. They certainly portend the future directions of word processing and its attendant corollaries, since obviously OCR, photocompositors, and intelligent printers can *easily* fit into such configurations right now! Small business computers, personal computers, and other desktop "intelligent terminals" are also being interfaced as part of the distributed networks and will be even more so in the future. And with electronic equipment continuing to fashion drastically improving performance/price curves, the fully distributed, automated office can't be too far behind.

TYMSHARE, INC.
OFFICE AUTOMATION DIVISION

DON

ANDREWS

PRESENTS

TYMSHARE, INC.
OFFICE AUTOMATION DIVISION

RECENT AND FUTURE AUGMENT DEVELOPMENT PROJECTS

TYMSHARE, INC.
OFFICE AUTOMATION DIVISION

RELEASE SCHEME

* EXPERIMENTAL

* FIELD TEST

* USER

* PREVIOUS old

TYMSHARE, INC.
OFFICE AUTOMATION DIVISION

PROFILE FEATURES

EASY TO DEFINE FEATURES

ASSOCIATED WITH A SUBSYSTEM

USER INTERFACE IMPROVEMENTS IN FUTURE

TYMSHARE, INC.
OFFICE AUTOMATION DIVISION

BUGS FILES

FILE KEPT WITH RELEASE

NUMBER ASSIGNED

BUG FIXES/CHANGES CATEGORIZED

GENERATION OF "NEWS"

TYMSHARE, INC.
OFFICE AUTOMATION DIVISION

COMMAND SEQUENCES

EXTENSION OF PROCESS STRUCTURES

FRANKLY, A PROGRAMMING LANGUAGE

FLOW CONTROL--LOOPS, TESTS, CALLS, RETURNS

ARGUMENTS AND RESULTS

USER INTERACTION

DEBUGGING AIDS

TYMSHARE, INC.
OFFICE AUTOMATION DIVISION

CURRENT AND FUTURE DEVELOPMENTS TYMSHARE AUGMENT 1250 TERMINAL

NEW TECHNOLOGY HARDWARE

MORE MEMORY

15" DIAG SCREEN

LOCAL SETUP COMMANDS

WIDE AND LONG SCREEN

MULTIPLE SAVED SCREEN

VT100 EMULATION

OTHER FEATURES

TYMSHARE, INC.
OFFICE AUTOMATION DIVISION

BETTER PROGRAMS BUFFER MANAGEMENT

COMPATABLE WITH "OLD" SUBSYSTEMS

CONFIGURATION STEP FOR "NEW"

SHARED CODE

SEMI-AUTOMATIC LOAD/UNLOAD

FASTER SUBSYSTEM LOADING

TYMSHARE, INC.
OFFICE AUTOMATION DIVISION

IDENT SYSTEM

NEW TECHNOLOGY

MULTIPLE IDENT DATABASES

REPRESENTS REAL WORLD

FEWER COLLISIONS

NEED IDENT PLUS SYSTEMNAME

ADMINISTRATION SUBSYSTEM

TYMSHARE, INC.
OFFICE AUTOMATION DIVISION

AUGMENT MAIL

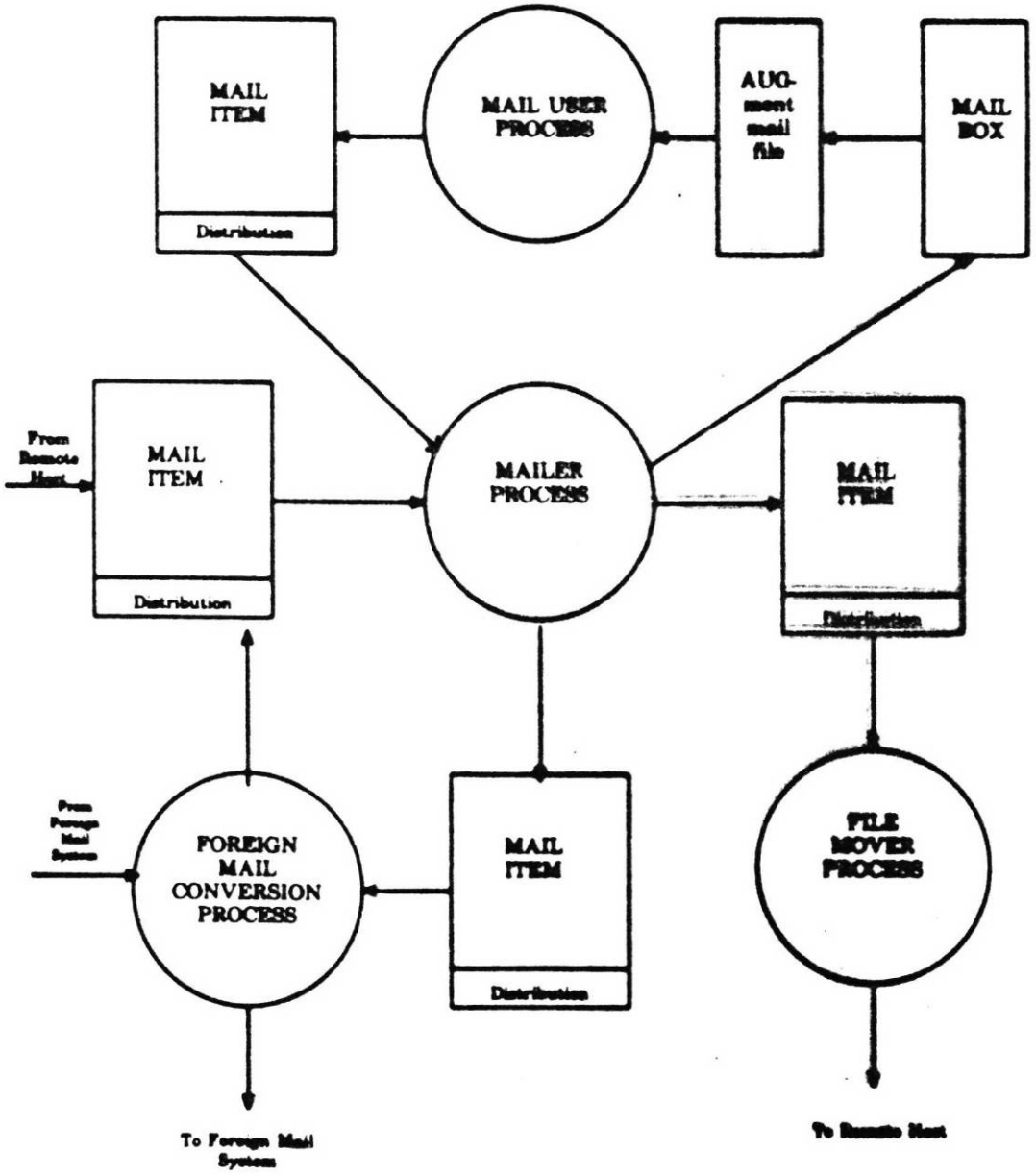
HISTORY

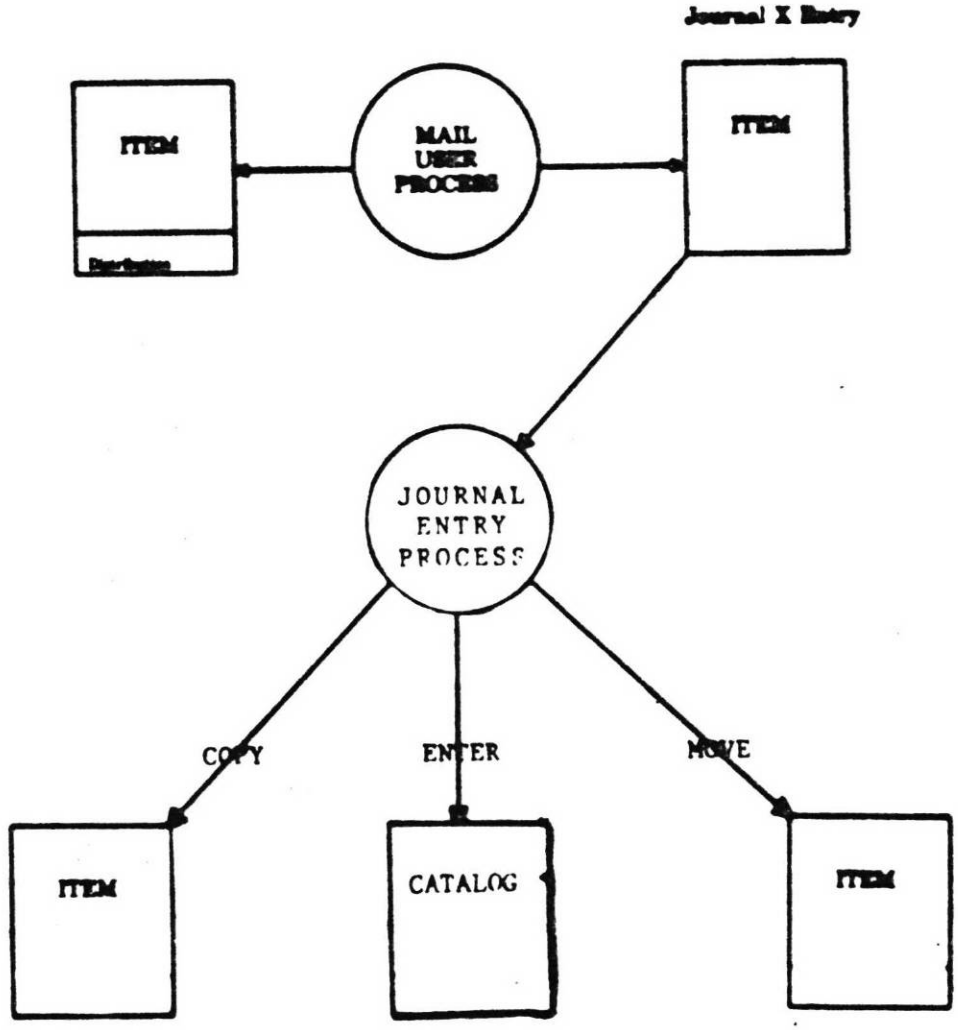
AUGMENT MAIL ITEM IS AUGMENT STRUCTURE

UNRECORDED/RECORDED

PUBLIC/PRIVATE

INTERFACES TO OTHER MAIL SYSTEMS


MAIL ITEM
Distribution
MAIL USER PROCESS
AUG-ment mail file
MAIL BOX
From Remote Host
MAIL ITEM
Distribution
MAILER PROCESS
MAIL ITEM
Distribution
FOREIGN MAIL CONVERSION PROCESS
MAIL ITEM
Distribution
FILE MOVER PROCESS
To Foreign Mail System
To Remote Host


Journal X Entry
ITEM
MAIL
USER
PROCESS
ITEM
JOURNAL
ENTRY
PROCESS
COPY
ENTER
MOVE
ITEM
CATALOG
ITEM

TYMSHARE, INC.
OFFICE AUTOMATION DIVISION

THE

END

University where he held the rank of Associate Professor until 1973. In June of 1973 he was appointed Professor and Head of the Research Group on Operating Systems I at the Technical University of Darmstadt, Germany, where he remained through August 1976. He is presently Professor in the Department of Computer Science, University of North Carolina, Chapel Hill. He is also with the Information Systems Staff, Communications Sciences Division, at the Naval Research Laboratory, Washington, DC. He has published papers in the areas of computer design languages and simulation techniques. His current interests are in the field of software engineering methods, computer system design, abstract specification for programs, verification that a program meets its specifications, and cooperating sequential processes.

The Design of a Message Switching System: An Application and Evaluation of Modula

GREGORY R. ANDREWS

Abstract—Modula is a new programming language for implementing dedicated, parallel systems. Following a systematic design technique, this paper illustrates the use of Modula for the design of a message switching communication system. A message switching system poses a number of interesting problems: a high degree of concurrent activity exists, a variety of I/O devices need to be controlled, messages can have multiple destinations, and messages can be preempted. The strengths and weaknesses of Modula with respect to these specific problems and its utility as a general purpose language are evaluated.

Index Terms—Concurrent systems, message switching, Modula, modular design, monitors, processes, software design, structured multiprogramming.

I. Introduction

THE unmistakable trend in recent years has been toward the use of high-level languages for systems programming. In an effort to improve upon available tools, three new languages have recently been designed: Concurrent Pascal [2] and Modula [8] aid in the design and implementation of multiprogramming systems while Euclid [4] is intended for implementing verifiable systems such as compilers or operating system nuclei. All three borrow heavily from the work of Wirth in the design of Pascal [7]. Although intended primarily for the development of small operating systems, both Concurrent Pascal and Modula are applicable to parallel systems in general.

In this paper, the design of one specific example, a message switching communication system, is developed using Modula. Our purposes are: 1) to present a system design technique; 2) to illustrate the use of Modula as a design, documentation, and implementation language; and 3) to evaluate Modula's utility. A message switching communication system was chosen as the application because it shows the range of Modula's applicability and presents a number of interesting implementation problems. Modula was chosen as the target language because it is specifically intended for dedicated multiprogramming systems, provides much needed facilities for controlling input/output, and appears to be very efficiently implemented [9], [10].

The specific design technique used here is described in Section II. The communication system itself is then developed in Sections III–V. Section III specifies the functions and external interfaces of the system. Section IV summarizes the major features of Modula and presents the system organization, information and control flow, and block interfaces in terms of Modula components. Section V refines the organization by giving outlines of Modula programs for the most interesting parts of the system. Finally, Section VI evaluates the utility of Modula for the design of parallel systems by reflecting on aspects of the communication system.

Modula is used here as a specification language since its compiler is not generally available. Even without a compiler though, we feel that programming any system in a structured, high-level language such as Modula is a valuable prelude to actual implementation. It serves as an intermediate step between specification and coding that helps one develop and reason about the implementation. It also provides meaningful documentation when used as comments in whatever implementation language is eventually employed.

II. Design Technique

The design described here was developed in three major steps: system specification, system organization, and program implementation. Each major step consists of a number of parts. The first step involves specifying the major functions of the system and the specific formats of input/output messages. This characterizes both what and how information is processed; hence it completely characterizes the user's

Manuscript received March 2, 1978; revised June 26, 1978. This work was supported in part by the Battelle Columbus Laboratories under Scientific Services Program Contract 0562 and in part under National Science Foundation Grant MCS 77-07554.

The author is with the Department of Computer Science, Cornell University, Ithaca, NY 14853.

0098-5589/79/0300-0138$00.75

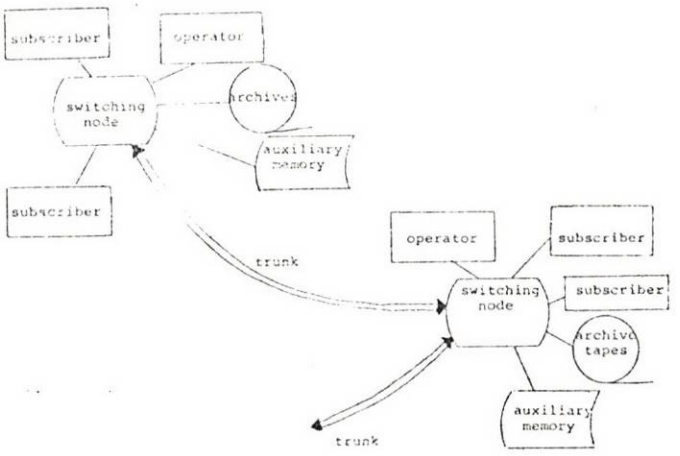
iew of the system. My role, as the designer, was to discuss ystem functions with intended users in order to understand nd characterize the purpose and scope of the proposed ystem. In addition, it is helpful to lay out a representative ardware configuration in order to identify the characteristics nd constraints of the implementation environment.

The second step, system organization, involves successive efinement of the system functions into a program structure terms of Modula constructs. Since Modula provides processes and modules [8] as its building blocks for parallel rograms, various organizations in terms of processes and odules were considered. First, important groups of processes and modules, corresponding to groups of similar I/O evices and to major system functions, were identified. Secnd, the types of information flowing between the groups as defined and the connection of the groups was sketched. hird, each group was refined into a specific organization. his involved outlining the actions of each process and identifying the operations (procedures) for each module. Finally, he interconnection of the processes and modules was specied and the paths and order of information and control low through the system were traced. These ordered lists of ctions proved very useful when the organization was deribed to others.

At this point the design was (and should be) discussed in etail with the people purchasing the system. This allowed isconceptions and ambiguities about system functions to e clarified. It helped the designer to be sure of the direction which he was headed and it helped the purchasers to better nderstand the program they would (hopefully) receive.

The final step is the implementation of each component. mplementation does not mean "start coding," however. irst, global data types were declared. Second, the procedures nd parameters for each interface module were precisely dened and comments describing the function of each procedure ere written. Third, comment and code outlines of each rocess were developed. (These three tasks involved refining e organization developed in the second step of the design.) inally, each component was programmed by first defining l variables and then writing language statements. Once the ganization was well understood and all variables were ecified, it became fairly easy to program each component. his is not to say that creativity was no longer needed!)

Few designs could proceed in a straight line through the bove steps and the one described here did not. Some itera- on occurred within and to a lesser extent between the second nd third steps (the first step, system specification, was fixed efore the program design began since it was independent of e implementation language). Within the second step, various ays to pass messages through the system were considered; ch organization involved different process/module organizaons and interfaces. The organization selected (and presented Section IV) appeared to be clearer and more efficient than e ones rejected. In the third step, slight changes to parameter lists and global data types were required as the processes nd modules were coded. Aside from one simple change to e system organization resulting from a misuse of Modula alling one interface module from within another), *no* change



subscriber:	full duplex terminal (up to 50 per node)
operator:	full duplex terminal
switching node:	processor(s)
archives:	tapes
auxiliary memory:	drum or disk
trunk:	high speed communication lines(s)

Fig. 1. Representative system configuration.

to one component affected anything more than parameter lists or global data types. This important benefit resulted from using a structured design approach and structured implementation language. This point will be returned to in Section VI after the design has been presented.

III. Specification of a Message Switching Communication System

The message switching system considered here is modeled on several communication systems currently employed by the U.S. government and NATO. It consists of a network of switching nodes connected via trunk lines. Locally attached to each switching node are up to fifty subscribers, an operator, archive tapes, and auxiliary memory. The function of each node is to route input messages to one or more output destinations. Input is received from local subscribers or from another switching node (via a trunk line), stored on auxiliary memory and then forwarded to output destinations, which can be either local subscribers or other switching nodes. Because messages must be completely received before being forwarded, this type of communication is often called store-and-forward message switching. A representative configuration, also indicating the type of hardware typically employed, is shown in Fig. 1. Since each switching node performs the same functions, we will focus our attention on one node.

Three successive phases are involved in processing each message: input, switch, and output. The input phase consists of reading input from a subscriber or trunk line and storing the message on both auxiliary memory and an archive tape. Each input message contains a header, body, and end marker as shown in Fig. 2. The size in the header is used as a maximal claim for auxiliary memory allocation and deadlock prevention.

In the switch phase, the header is examined to determine

```
header:  start of header code
         message identifier (origin, date, and time)
         size (omitted for subscriber output)
         destinations (omitted for subscriber output)
         priority
         [other data not of concern here]
         end of header code

body:    sequence of characters (in 80 character blocks for trunk lines)

end:     end of message code
```

Fig. 2. Subscriber and trunk message formats.

the output destinations. For each destination, a directory is consulted to determine the appropriate output line to use (either local subscriber or trunk to a remote destination) and a copy of the message is queued for output on each distinct line. Each destination eventually receives its own copy.

In the output phase, a message is retrieved from auxiliary memory and written on the appropriate output line in the format shown in Fig. 2. Each message contains a priority as part of its header so at all times the highest priority message for an output line is transmitted. This means that a partially written message can be preempted by a newly queued one. If preempted, a message is later retransmitted in its entirety.

Each switching node has one operator who can send and receive messages like any subscriber. In addition, the operator monitors and controls the node's activity. The operator can request certain actions (e.g., cancel a message) and is notified when exceptions are detected (e.g., end of archive tape). Although necessary and important in an actual communication system, the operator will be ignored here because the main function of the switching node is to process input messages.

In order to efficiently implement this communication system, four interesting problems need to be solved.

1) Maximal I/O parallelism must be provided so that each subscriber and trunk is kept as busy as possible.

2) Two different types of I/O devices, trunks and terminals, need to be controlled and yet both process similar messages.

3) The switch phase needs to coordinate the activity of all lines; in particular, output to multiple destinations must be coordinated.

4) An output message can be preempted, hence output controllers need to have a simple, efficient way to detect that preemption should occur; they also need to be able to restart transmission of a preempted message.

We now turn our attention to the design of a Modula implementation that solves each of these problems.

IV. System Organization in Modula

For sequential programming, Modula contains a set of data types, statements, and procedures based on those of Pascal, as well as a module construct. As building blocks for parallel programs, it also provides processes, interface modules, device modules, and signal variables. A process has the same structure as a procedure; namely, it has parameters, local variables,

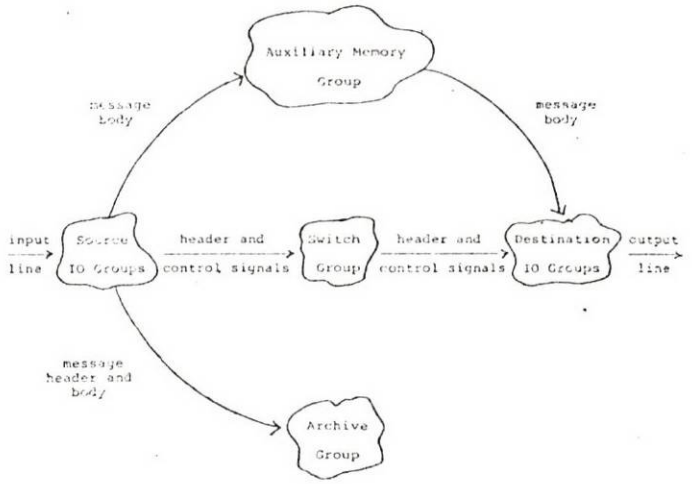


Fig. 3. System components and message flow.

and a sequence of statements. The fundamental difference is that when a process is "called" during system initialization, it executes concurrently with other processes. Processes interact via global variables and modules; except for device driver processes (see below) they can only be declared at the outermost program level.

Modules are like blocks in the Algol sense (they contain variables and statements). The difference is that a module forms a barrier between the objects it declares and those global to it. Its purpose is to implement an abstraction such as a message queue. It does so by selectively defining those objects that represent an intended abstraction while hiding those objects involved in its representation. Two special types of modules play a key role for multiprogramming: interface modules and device modules. Interface modules, which correspond to monitors [3], [3], guarantee mutual exclusion of defined procedures and provide for process synchronization via signal variables. Device modules are interface modules that contain device driver processes. There is one driver process in a Modula program for each addressable I/O device. Each driver activates its associated device, waits for device completion via a special doIO statement, and interacts with other processes via device module variables.

A. System Components

As described in Section II, each message processed by a switching node goes through three phases: input, switch, and output. To implement these phases, we need groups of processes and modules to control I/O lines, provide temporary storage for messages, provide an archive, and schedule output. The groups transmit message headers and message bodies. This leads to the organization shown in Fig. 3. Each input/output group manages one subscriber or trunk line. The auxiliary memory group provides a file system. The archive provides a tape storage system for recording all node activity. The switch group coordinates activity. As illustrated in Fig. 3, during input a message body flows from an I/O group to an auxiliary memory file; during output it flows from the file to the destination I/O group(s). Message headers and other control information flow through the switch group.

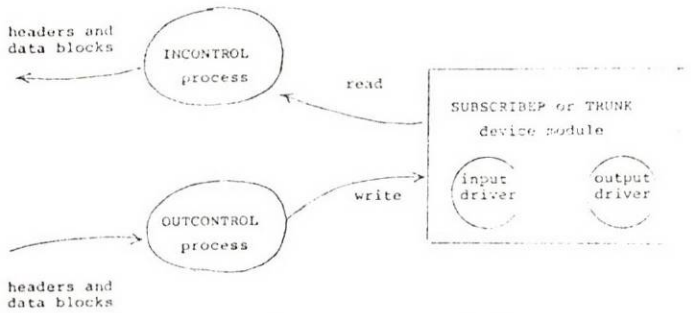


Fig. 4. Input/output group organization.

now refine this global organization by defining the spe- c actions and internal organization of the I/O, auxiliary mory, and switch groups. The archive is not considered ther; it is a straightforward addition.

nce subscriber and trunk lines are full duplex, we need have both input and output driver processes for each . Although they could be in separate device modules, thetically it makes sense to declare one device module ing two internal drivers. Two major activities take place ing input (output): formatting (or unblocking) messages, the actual reading (writing). The first is related to the *lication* whereas the second is related to the *hardware*. refore they should be logically separated in case either nges. The final I/O consideration is that within the switch- node it is more efficient to transmit data a block rather a character at a time. Given these considerations, the sen organization for input/output groups is shown in 4. Each INCONTROL process reads characters, and parses n into blocks of a message. Each OUTCONTROL process s the reverse. Each SUBSCRIBER or TRUNK device module ies about the peculiarities of its device. Note that even gh terminals and trunks are themselves quite different, software organization of both groups is *identical*. Only nal details differ.

e auxiliary memory group provides temporary storage messages. In particular, it provides INCONTROL and CONTROL processes with sequential files. In addition, auxiliary memory group must efficiently control the age device. It therefore has two distinct levels: a file em level, which provides file operations and manages space; and a device level, which schedules and performs and manages sector buffers. This leads to the organization wn in Fig. 5. MEMORY defines five operations that provide bstract file system. Within MEMORY there is the AUXMEM ce module that implements the abstraction. Note that use AUXMEM is wholly inside MEMORY, no process global MEMORY can directly access AUXMEM; global processes only see and need only be aware of the file operations.

nce there is a need for one central controller, the switch p consists of one process, SWITCH, together with interface ules that implement various services. The function of CH is to coordinate input and output; hence it needs talk to every INCONTROL and OUTCONTROL process. also needs to talk to the operator; this is being ignored

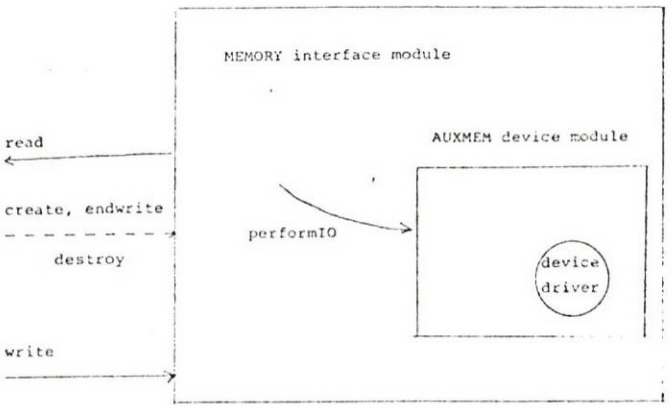


Fig. 5. Auxiliary memory group organization.

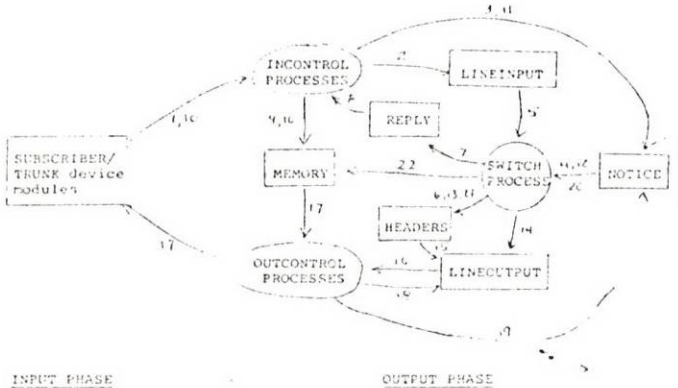
here.) INCONTROL and OUTCONTROL processes signal SWITCH whenever there is work for it to do. Because it can only wait within one interface module at a time, SWITCH receives all its work from NOTICE. Each notice specifies a kind and, optionally, other parameters. The three major kinds of notices are "start of input," "end of input," and "done." A "start of input" notice causes SWITCH to receive a header from LINEINPUT, which it then stores in HEADERS; SWITCH then gives a REPLY back to the appropriate INCONTROL process. (INCONTROL waits for a REPLY before proceeding because SWITCH may need to delay starting new input if the node is heavily congested.) An "end of input" notice causes SWITCH to insert one output command in LINEOUTPUT for each output line indicated by the destinations field in the header of the completed input message. The "done" notice signals completion of output to one destination; when all destinations are done, SWITCH deletes the saved header from HEADERS and destroys the MEMORY file containing the message. The operations provided by each interface module in the switch group are shown in Fig. 6.

B. Component Connections and Information Flow

We can now put the three main groups—input/output, auxiliary memory, and switch—together. Fig. 7 shows the actions taken in processing an entire message and the order in which they occur. The arrow on each arc indicates the direction of flow of parameters. (Within each interface module signal variables are used to synchronize processes.)

```
NOTICE
        post
        receivenotice

LINEINPUT
        sendhead
        receivehead

HEADERS
        enterheader
        retrieveheader
        updateheader
        deleteheader

REPLY
        give
        receivereply

LINEOUTPUT
        insertoutput
        receiveoutput
        doneoutput
```

Fig. 6. Switch group–interface module operations.



INPUT PHASE

1. read header contents
2. send header
3. post "new input" notice
4. receive "new input" notice
5. receive header
6. store header
7. send reply
8. receive reply
9. create file and write header
10. read blocks and write to file
11. send "end of input" notice

SWITCH PHASE

12. receive "end of input" notice
13. update header
14. send output commands - one per destination

OUTPUT PHASE

15. retrieve header
16. receive output command
17. read file and write on line
18. call done operation
19. send "done" notice

TERMINATION

20. receive "done" notices - one per destination
21. delete header
22. destroy file

Fig. 7. Component interfaces and timing.

Although not shown in Fig. 7, the operator communicates with SWITCH and MEMORY in the same manner as regular subscribers. Operator requests are sent to switch as special kinds of NOTICES and exceptions are sent from SWITCH to the operator via LINEOUTPUT. In a real application, the operator also communicates with a retrieval process in order to fetch old messages from archive tapes. These details are shown elsewhere [1].

V. System Implementation–Program Outlines

A Modula program is a module. In our case, the MSG . SW module contains declarations of data types common to all components; device modules, interface modules and processes for each component in Fig. 7; and code that initializes tables and activates the processes. Programs for the major components are outlined here in the manner in which they were designed (as described in Section II). Since our main goals are to illustrate and evaluate the use of Modula, we do not show the actual variable declarations and statements (the last step of the design process). Complete programs are given in [1].

A. Global Data Types

The INCONTROL, OUTCONTROL, and SWITCH processes exchange several types of information. The two most important kinds are message headers and data blocks. To facilitate the declaration of variables referring to these types of information and to make use of Modula's type checking facilities, the format of header and block are specified using global type declarations. In particular, header is declared as[1]

```
type header = record
        identifier : record
                origin, date, time : integer end;
        size : integer;
        outputcount : integer; (*number of destinations*)
        destinations : array 1 : "max" of integer;
        .
        .
        .
        end
```

A block is simply an array of characters:

```
type block = array 1 : "blocklength" of char
```

Other types of variables for communicating with the archive module and operator are also needed in practice.

B. Device Modules

The SUBSCRIBER and TRUNK device modules provide interfaces to terminals and trunk lines, respectively. Each defines two procedures, read and write, that perform buffer management. Each also contains two driver processes, one for input and one for output, that perform actual I/O and synchronize with the procedures. An outline of a SUBSCRIBER device module is shown in Fig. 8. Character buffers are used for I/O; counters, pointers, and signals are used for synchronization. Note that one SUBSCRIBER device module must be declared for each terminal. There is no way in Modula to declare a type of device module and subsequently declare as many instances as are required.

Each TRUNK module has the form shown in Fig. 9. A TRUNK differs from a SUBSCRIBER in that trunks transmit

[1] As a coding convention, constants are denoted by names enclosed by quotes. They would in practice be declared as constants.

```
device module SUBSCRIBER;          (*one per terminal*)

    define      subread, subwrite;

    var         (*buffers, counters, and signals*);

    procedure   subread (var ch: char);
                (*retrieve next character from input buffer when available*)
                end subread;

    procedure   subwrite (ch: char);
                (*deposit ch in output buffer and signal output*)
                end subwrite;

    process     input;
                (*input characters as long as input buffer is not full*)
                end input;

    process     output;
                (*output characters as long as output buffer is not empty*)
                end output;

    begin       initialize variables; input; output

    end SUBSCRIBER
```

Fig. 8. Subscriber device module.

```
device module TRUNK;          (*one per trunk line*)

    define      trunkread, trunkwrite, writecontrol, postcontrol,
                waitcontrol;

    use         block;

    var         (*buffers, counters, and signals*);

    (*procedures:
        trunkread    - get next block from input buffer
        trunkwrite   - deposit block in output buffer
        writecontrol - deposit control character in output buffer
        postcontrol  - save control character
        waitcontrol  - retrieve posted control character*)

    process     input;
                (*input next character(s) into buffer and determine
                  type of character*)
                end input;

    process     output;
                (*output next control character or data block*)
                end output;

    begin       initialize variables; input; output

    end TRUNK
```

Fig. 9. Trunk device module.

blocks instead of single characters and numerous control characters are transmitted to synchronize the trunk lines. Since trunks connect switching nodes, there would be a TRUNK module in the node at each end of the line; information output from one TRUNK module is read as input by the other. The writecontrol procedure is used to output a control character. Waitcontrol is used to wait for a control character response from prior output of a data block. The response is sent by the switching node at the other end of the trunk line; when received (via trunkread) its presence and value are recorded via postcontrol.

C. *Interface Modules*

Six interface modules implement communication paths between the processes: MEMORY, LINEINPUT, NOTICE, REPLY, HEADERS, and LINEOUTPUT. MEMORY and LINEOUTPUT are both quite interesting; the others are straightforward.

```
interface module MEMORY;

    define      createf, writef, endwritef, readf, destroyf;

    use         block;
    (*variables - directory, free space, synchronization*)
    (*utility procedures - spaceavail, request, release- for managing
                                                        free space*)

    procedure   createf (msgid, size : integer; var filename : integer);
                (*create file to store up to size blocks of msgid;
                  return name in filename*)

    procedure   writef (filename : integer; buffer : block);
                (*store buffer as next record in filename*)

    procedure   endwritef (filename : integer);
                (*release space allocated to but not used by filename*)

    procedure   readf (filename, blocknumber : integer; var buffer : block);
                (*fetch the record given by blocknumber from filename
                  and store it in buffer*)

    procedure   destroy (filename : integer);
                (*destroy filename and release its space*)

    device module AUXMEM;

        define      performIO; use block;
        (*variables - sectorbuffer(s), synchronization*)
        (*scheduling procedures - requestturn, releaseturn*)

        procedure   performIO (operation, sector, offset : integer;
                               var buffer : block);
                    (*schedule IO and then signal IO*)

        process     driver;
                    (*wait for IO; perform operation; signal done*)

        begin       (*initialize*)

        end AUXMEM;

    begin  (*initialize free space etc.*)

end MEMORY;
```

Fig. 10. Memory interface module.

The MEMORY module defines five operations for managing files: createf, writef, endwritef, readf, and destroyf. Its program is outlined in Fig. 10. Createf is called by an INCONTROL process; it passes a message identifier and maximum file size as parameters. Createf allocates (actually it just commits) adequate space, waiting if necessary until space is available. A directory entry is then initialized and a filename is assigned and returned to the calling process. The INCONTROL process subsequently fills the file by calling writef, specifying the filename and data block (of type block declared earlier). The writef procedure stores the block in the next allocated external record by calling AUXMEM. A file is "closed" by calling endwritef; its purpose is to allow MEMORY to deallocate any space originally requested (via createf) but not actually used.

Message files are read by OUTCONTROL processes by calling readf, specifying the filename, block (record) number, and storage variable used for returning the block. MEMORY maps the filename and block number into a sector address and offset and calls AUXMEM to perform the read. The block number must be specified on calls to readf because many OUTCONTROL processes may simultaneously be reading the same file (messages may have multiple destinations). Once all file reading is completed, SWITCH calls destroyf, which frees the space occupied by the file and, if necessary and possible, awakens a process waiting within createf.

AUXMEM is a device module within MEMORY that schedules

and performs read and write operations on auxiliary memory. AUXMEM defines one procedure, perform I/O, that takes an operation, sector, offset, and buffer and performs the operation (read or write). Perform I/O requests a turn on the device, synchronizes with the device driver, then releases its turn. Device scheduling is handled by internal procedures that implement a scheduling discipline appropriate for the actual type of device used. Notice that many processes could be waiting within AUXMEM. Only one at a time can be waiting for the driver to complete, though. While all are waiting and the device is servicing a request, future requests for file access can be handled by MEMORY and queued for scheduling within AUXMEM. The exclusion and synchronization mechanisms of Modula's interface and device modules make this possible.

The other interesting interface module is LINEOUTPUT. It schedules and controls output activity via three operations: insertoutput, receiveoutput, and doneoutput. For each output line (trunk or subscriber), LINEOUTPUT maintains a list of output messages where each list element identifies a message by its HEADERS index and priority. Each output list is ordered by priority. The insertoutput operation is called by SWITCH, once for each output destination; it inserts the new message in the appropriate place in the output list. If the new message is of higher priority than the one previously at the head of the list, a locally declared but exported (via define) preempt flag is set. This will cause the appropriate OUTCONTROL process to stop working on what it is doing and call receiveoutput to fetch the new, higher priority output message. Each OUTCONTROL process is therefore always working on the message at the head of its output list.

Receiveoutput is called by each OUTCONTROL process either when it is ready to output a new message or when it has found the preempt flag set. If no messages are available, the OUTCONTROL process waits for one to be inserted. When one is available, the header is retrieved from HEADERS and returned to the OUTCONTROL process. Receiveoutput leaves the output message on the output list so that it can be received again if it gets preempted. Once output is complete, an OUTCONTROL process calls doneoutput, which deletes the appropriate entry from the process' output list.

The other four interface modules connect INCONTROL and OUTCONTROL processes to the SWITCH process. LINEINPUT is a simple message passing module that has two operations, sendheader and receiveheader, used to pass headers from INCONTROL processes to SWITCH. NOTICE implements a bounded buffer of notices for SWITCH. It has two operations, post and receivenotice, that synchronize with each other in the usual, bounded-buffer fashion (see [3] for example). REPLY is similar to NOTICE and has two operations, give and receivereply. All headers of active messages are stored in HEADERS by calling the enterheader operation. The header can be subsequently retrieved, updated, and deleted by calling the other three HEADERS operations.

The operations and parameters for each interface module are summarized in Fig. 11. The comment with each operation summarizes its role and actions.

```
LINEINPUT -
    procedure  sendhead (hd : header);
               (*save header*)
    procedure  receivehead (var hd : header);
               (*retrieve next header*)
NOTICE -
    procedure  post (kind, data : integer);
               (*save kind and data; signal receivenotice*)
    procedure  receivenotice (var kind, data : integer);
               (*fetch next notice when available*)
REPLY -
    procedure  give (linenumber, data : integer);
               (*save data for INCONTROL process of linenumber*)
    procedure  receivereply (linenumber : integer; var data : integer);
               (*retrieve data saved for linenumber when available*)
HEADERS -
    procedure  enterheader (hd : header; var index : integer);
               (*save header; return location in index*)
    procedure  retrieveheader (var hd : header; index : integer);
               (*return copy of header at location index*)
    procedure  updateheader (hd : header; index : integer);
               (*store hd at location index*)
    procedure  deleteheader (index : integer);
               (*delete header at location index*)
LINEOUTPUT -
    procedure  insertoutput (line, msgindex; priority : integer);
               (*store (msgindex,priority) at specified priority on
                 output list of line; set pre-empt flag if necessary*)
    procedure  receiveoutput (line : integer; var hd : header);
               (*return header of first message on output list of line
                 as soon as one is available*)
    procedure  doneoutput (line : integer)
               (*delete first entry from output list of line*)
```

Fig. 11. Summary of interface module operations.

```
process  SUB_INCONTROL; (*one per SUBSCRIBER device module*)
         (*variables for block, header, character, status, etc.*)
    begin  status := "findstart";
    loop
        subread(character);  (*get next input char*)
        case status of
        "findstart" : begin
                      (*look for start of message sequence*)
                      if start of message then status := "inhead"
                      end;

        "inhead" : begin
                   (*store character in appropriate header spot*)
                   if end of header then
                      sendhead(header); post ("start of input");
                      receivereply( );  (*from REPLY*)
                      createf( );       (*MEMORY file*)
                      status := "in body"
                      end
                   end;

        "in body" : begin
                    (*store character in block*)
                    if end of input message then
                       writef( );              (*store last block*)
                       endwritef( );           (*in MEMORY*)
                       post("end of input");   (*NOTICE*)
                    elseif end of block then writef( )
                       end
                    end
        end (*of case*)
    end (*of loop*)
end SUB_INCONTROL
```

Fig. 12. Subscriber input control process.

D. Processes

In this section, outlines of the three kinds of message switching system processes are given. INCONTROL processes read from device modules and build input messages. Because there are two types of devices in out prototype system, there are two types of INCONTROL processes. The outline for a subscriber input controller is shown in Fig. 12. It reads characters one at a time and then takes certain actions depending on where it is in processing an input message. The three cases in Fig. 12 illustrate the three main actions. Ex-

```
process  SUB_OUTCONTROL;  (*one per SUBSCRIBER device module*)
         (*variables for header, block, and counters*)
 begin (*initialize variables*)
 loop
    loop  (*inner loop is executed once for each output message--
           outer loop allows escape from inner loop when preemp-
           tion occurs*)
       receiveoutput (line, hd);  (*from LINEOUTPUT*)
       (*format output message header*)
       repeat  (*for each header character*)
               subwrite (character)
          until  end of header;
       when pre-empt [line] do exit;
       repeat  (*for each data block of output message*)
          readf( );  (*from MEMORY*)
          repeat  (*for each block character*)
                  subwrite (character);
             until  no more characters;
          until   no more blocks or pre-empt [line];
       when pre-empt [line] do exit;
       doneoutput (line);  (*to LINEOUTPUT*)
       post ("done")  (*to SWITCH*)
       end  (*of inner loop*)
    end (*of outer loop*)
end  SUB_CONTROL
```

Fig. 13. Subscriber output control process.

ceptions, such as an error or cancellation, are not illustrated, however. (In practice, exceptions detected by control processes result in a NOTICE being posted for SWITCH, which in turn informs the appropriate subscriber and/or the operator by using LINEOUTPUT.)

Trunk input controllers are similar to subscriber input controllers. They use the interface modules of Section V-C in the same way and take basically the same actions. The differences are that trunks transmit blocks instead of characters and require numerous control characters. The main point of interest is that the difference between the trunk and subscriber controllers results from, and only affects, the associated device module. The remainder of the switching system (MEMORY, SWITCH, etc.) is unaware of the difference.

OUTCONTROL processes transmit output messages from MEMORY to device modules. Again there are two types of output controllers, one for each type of output line. An outline of a subscriber output control process is shown in Fig. 13. A trunk output controller is similar, differing only in the ways described above for the input controllers. Each SUB_OUTCONTROL process executes a loop, once for each output message whose header is received from LINEOUTPUT. It first formats and writes an output header. It then transmits the body of the message a block at a time. Each block is read from MEMORY and then written to the appropriate device module. On completion of output, LINEOUTPUT's doneoutput operation is called, SWITCH is notified, and the output process loops to receive the next message when it is available.

The most interesting aspect of output controllers is the way they handle preemption. Because a new output message of higher priority may become available while SUB_OUTCONTROL is busy with another message, it must detect the need for preemption and be able to stop what it is doing. Each output controller periodically checks the appropriate preempt flag exported from LINEOUTPUT and, if set, exits the inner loop statement via the Modula when statement. As shown in Fig. 13, the flag is checked just after each block of data is outputted.[2] This could readily be changed to character level checking merely by adding "or preempt [line]" to the until clauses of the first and third (innermost) repeat statements.

The final system component is the SWITCH process, which directs all activity. It is outlined in Fig. 14. All communication to SWITCH is via the NOTICE interface module. For each notice, SWITCH executes a case statement as shown. A "start of input" notice signals the presence of a new input header which is received from LINEINPUT and stored in HEADERS. An "end of input" notice causes SWITCH to generate an output command for each destination named in the message's header. A "done" notice signals output completion for one destination; once all destinations have sent "done" notices, the message header and file are destroyed. In an actual system, other notices will also be handled by

[2] Because the when statement must appear on the same level as other statements in a loop, repeat statements must first be terminated before the loop can be exited via when.

```
process SWITCH;
        (*variables for names of active messages (HEADERS indices),
          output directory, header, kind and data from NOTICE,etc.*)
    begin  (*initialize variables*)
    loop   receivenotice(kind, data);
           case kind of
       "start of input" : begin
                receivehead(head);  (*from LINEINPUT*)
                enterheader(head,index);  (*to HEADERS*)
                (*store index in active names, etc.*)
                give( )   (*REPLY*)
                    end;
       "end of input" : begin
                retrieveheader(head, index);
                repeat  (*for each unique destination line*)
                      insertoutput( );  (*in LINEOUTPUT*)
                      until no more destinations
                end;
       "done" : begin
                decrease output count;
                if output complete then
                     deleteheader( );
                     destroyf
                     end
                end
       (*other cases for exceptions, cancel, etc.*)
       end  (*of case*)
    end (*of loop*)
end  SWITCH
```

Fig. 14. SWITCH process.

SWITCH, for example to stop a line, cancel a message, or process an error.

VI. Evaluation of Modula

The design of the message switching communication system has been described here in the order in which it was developed. We feel that this three stage design process (system specification, system organization, and program implementation) represents a valuable, structured approach to the design of software systems. We also feel that Modula is a very good language for dedicated multiprogramming systems such as the message switch. To be specific, Modula is good in four major respects.

First, the major new building blocks of Modula—processes and modules—were both appropriate and easy to use. One of the easiest ways to tell if a language is suited to the problem is to see whether it helps or hinders the refinement of system functions into a program implementation. A good target language should guide the organization and implementation by providing a structured framework and adequate tools. As mentioned in Section II, very little iteration occurred as the design was developed. This resulted from the fact that system functions mapped readily into processes and modules. Although one might have wished to have one process for each input message, which is not possible in Modula since there must be a fixed number of processes created only during system initialization, we feel that the chosen organization is superior because it makes it easier to provide simultaneous output of one message at multiple destinations and to implement output preemption. Modules also worked well for this problem: interface modules made it easy to implement process communication and synchronization, and device modules provided a nicely encapsulated means for implementing device interfaces. Both served their intended purpose of information hiding since no choices about data representation or coding in one module affected anything more than the parameters passed by processes that invoked it.

The second positive aspect of Modula is the utility of device modules as a means for representing devices. It is the device module concept more than any other that, in our opinion, makes Modula the most attractive multiprogramming language available today. A device has much in common with any abstract data type: it accesses storage, defines operations, and requires exclusion and synchronization. Device modules, as a special kind of interface module, provide all these facilities. They give user processes a procedural interface to devices, implicitly provide exclusion, have signal variables for synchronization, and have driver processes to achieve I/O parallelism. They also allow details of device handling, such as buffer management, to be hidden from the user. In fact, Modula makes it possible to cleanly separate the file system functions in MEMORY from the device functions in AUXMEM. Finally, the device module construct makes it possible to control two quite different types of I/O devices, subscribers and trunks, while providing a very similar interface to the controller processes.

The third major benefit of Modula is the ease and efficiency with which output preemption is implemented. The preemption problem occurs in many parallel systems so a multiprogramming language must make it easy to handle. The specific Modula feature of value here is the ability to export read-only variables from modules. The OUTCONTROL processes test for preemption merely by checking a flag; if this flag were not accessible outside of LINEOUTPUT, output controllers would have to call a LINEOUTPUT procedure each time they wanted to check a flag. Also note that preempted messages could be easily restarted by keeping them on the output list in LINEOUTPUT.

The final positive aspect of Modula is its apparent execution efficiency. The storage and time required by the Modula kernel is minimal [10]. Consequently most of the space and time used by the message switching or any other system should result from functions of the system itself.

Despite its power, Modula is deficient in four respects as a general language for the design and implementation of parallel systems. It is intended for small, static systems that reside entirely in main memory. As such it does not provide tools for implementing large systems (such as contemporary time-sharing systems) requiring virtual memory, dynamic creation and destruction of processes, quantum scheduling, executable files, dynamic access control, or combinations of exclusive and concurrent access to shared variables. Modula processes must be created during system initialization, regular (non-device) processes execute at the same priority, and all processes execute until they wait for a signal. Interface modules guarantee that at most one process at a time may execute within a defined procedure. This is not necessary, though, if processes only read shared data. For example, if the MEMORY module used direct access storage rather than

auxiliary storage, it would be desirable to allow several processes at a time to simultaneously read files, yet provide exclusive access for a process that updates a file. This is not possible using interface modules.

A second deficiency is that two features of Modula, exported variables and driver processes, create problems for verification because both allow shared variables to be altered while being read.[3] Exporting variables, as noted, made it possible to efficiently handle preemption. Driver processes in Modula have higher priority than regular processes and maintain control after executing a send operation on a signal variable so that they execute whenever possible. Both features increase execution efficiency but at the price of increased verification complexity. They are specific examples of a general, unresolved dilemma of programming language design. To verify that the use of a read-only variable is correct, it is necessary to show that assertions in "reader" processes do not interfere with those in the interface module that can change the variable. To verify the correctness of a device module, it is necessary to show that driver processes do not interfere with processes that may be executing within procedures in the same device module. Unfortunately, as becomes evident from a close reading of [6], it is not easy to prove that two processes are interference-free.

The third deficiency is that Modula does not allow processes or modules to appear in type declarations. As a result, one device module, input controller, and output controller must be declared for each subscriber line even though the actions of each are identical. The same problem occurs with trunks. As long as code is shared, this does not affect efficiency because distinct names and storage space are obviously required. The readability and clarity of a full listing of the system would be much worse, however. Concurrent Pascal, by using a different access control and activation scheme, does not suffer from this defect [2]. A related problem is the absence of generic types. Queues were used in the implementation of many of the interface modules. Even though identical except for their size and the type of information they contained, each queue had to be programmed separately.

The final deficiency of Modula as a general multiprogramming language is its method for specifying scope rules. Define and use allow each module to export and import what it wants. Since a module is used to provide a service, it is natural for it to define which objects are accessible to others. The use list, however, only protects against carelessness. If access security is required, a module should be told what it can use rather than take what it needs; it is then more appropriate for the declarer of an object to specify which other blocks can access it. In addition, facilities for dynamic access control are needed to implement, for example, a file system. The general problem of access control in parallel programs as well as two specific language proposals are described in [5].

The above four deficiencies must be removed in a language intended for general systems programming. They should not, however, obscure the facts that Modula is the most powerful multiprogramming language currently available and that it meets its stated goals. The message switching system described here is but one example of its utility.

[3]The previous paragraph referred to read-only simultaneous access; this does not cause a verification problem.

Acknowledgment

S. Amoroso and D. Morris proposed this application and served as sounding boards for the design as it was developed. A. Demers and D. Gries carefully read an earlier version of this paper. The referees also provided numerous, very constructive comments.

References

[1] G. R. Andrews, "Modula and the design of a message switching communication system," Dep., Comput. Sci., Cornell Univ., Ithaca, NY, Tech. Rep. 78-329, Jan. 1978.

[2] P. Brinch Hansen, "The programming language Concurrent Pascal," *IEEE Trans. Software Eng.*, vol. SE-1, pp. 199–207, June 1975.

[3] C. A. R. Hoare, "Monitors: An operating system structuring concept," *Commun. Ass. Comput. Mach.*, vol. 17, pp. 549–557, Oct. 1974.

[4] B. W. Lampson *et al.*, "Report on the programming language Euclid," *Sigplan Notices*, vol. 12, pp. 1–79, Feb. 1977.

[5] J. R. McGraw and G. R. Andrews, "Access control in parallel programs," *IEEE Trans. Software Eng.*, vol. SE-5, pp. 1-9, Jan. 1979; also Dep. Comput. Sci., Cornell Univ., Ithaca, NY, Tech. Rep. 78-328, Jan. 1978.

[6] S. Owicki and D. Gries, "An axiomatic proof technique for parallel programs I," *Acta Informatica*, vol. 6, pp. 319–340, 1976.

[7] N. Wirth, "The programming language Pascal," *Acta Informatica*, vol. I, pp. 35–63, 1971.

[8] —, "Modula: a language for modular multiprogramming," *Software—Practice and Experience*, vol. 7, pp. 3–35, 1977.

[9] —, "The use of Modula," *Software—Practice and Experience*, vol. 7, pp. 37–65, 1977.

[10] —, "Design and implementation of Modula," *Software—Practice and Experience*, vol. 7, pp. 67–84, 1977.

Gregory R. Andrews was born in Olympia, WA, in 1947. He received the B.S. degree in mathematics from Stanford University, Stanford, CA, in 1969 and the Ph.D. degree in computer science from the University of Washington, Seattle, WA, in 1974.

Since 1974 he has been an Assistant Professor of Computer Science at Cornell University, Ithaca, NY. Previously he worked for the Boeing Company, Seattle, WA. Since 1975 he has been a consultant to the Center for Tactical Computer Sciences, U.S. Army Electronics Command, Fort Monmouth, NJ. His interests include concurrent programming languages, operating systems, protection, and security.

Dr. Andrews is a member of the Association for Computing Machinery.

Information Systems News 11/12/79 p.37 col. 1

ACS America Develops 'Total Information' System

NEW YORK -- ACS America is developing an information processing system that will include word processing, electronic mail, automatic proofreading, message switching, interfacing for photocomposition and COM equipment, OCR, voice recognition and audio response.

Called PRIMACS, the system is based on ACS software and hardware from Prime Computer. The word processing and electronic mail modules are available now, while the total system will be complete in 1981, the firm said.

The system will be designed to operate with Prime's general-purpose minicomputers and, using standard Prime systems software, will be designed to be co-resident with any normal data processing functions.

One To 35 CPUs

PRIMACS will operate with one to 35 CPUs in a network, with each machine servicing up to 63 users simultaneously. ACS America installs the Primacs system on existing Prime hardware or furnishes both hardware andsoftware as a turnkey system.

The PRIMACS-I module is available now, at $15,000, and provides word processing capability, as well as permitting interfacing with other data processing systems and to the ACS Telex message-switching system.

It is menu-driven, allowing the operator to select functions that will access, create, print or delete a document.

PRIMACS-II is an extension of, and works in conjunction with, PRIMACS-I, and is also $15,000. It provides electronic-mail distribution of documents or administrative messages among system users on one CPU or among users of any CPU within the network.

This module also provides document filing, retrieval and archiving capabilities for each user and maintains users' diaries and schedules.

PRIMACS-III, scheduled to be available in December and priced at $10,000, contains a 60,000-word English-language dictionary and up to four foreign-language dictionaries.

It provides automatic hyphenation, document proofreading by comparing input word-by-word with the dictionary, key-word search, text indexing, system-generated tables of contents and sectional page and paragraph numbering.

Foreign-Language Function

PRIMACS-III is superimposed on PRIMACS-I and works with PRIMACS-II. The foreign-language dictionaries provide spelling checks and hyphenation in the same manner as the English dictionary. They will also permit a logical, word-for-word translation from and to a foreign language and English.

PRIMACS-IV, due in 1980, operates with PRIMACS I and II and provides automatic Telex and TWX message-switching functions for both incoming and outgoing messages.

Direct Output

PRIMACS-V, scheduled to be available six months after PRIMACS-IV, works with I, II and/or III and provides interfacing modules for direct output to automatic typesetting equipment and/or to computer output microfilm (COM) equipment.

PRIMACS-VI, due six months after PRIMACS-V, works with I, II, III and/or IV. It provides full-page OCR optical scanning input for editing, filing, retrieval and distribution.

PRIMACS-VII, due six months after PRIMACS-VI, works with all other PRIMACS modules and incorporates audio input based on voice recognition, plus audio response using digitized text readout techniques.

The PRIMACS system, written in FORTRAN IV, requires one Diablo letter-quality printer and either an Ontel OP1/R-ACS intelligent terminal or a Perkin-Elmer Owl 1200 non-intelligent CRT with 16 function keys. Both terminals may be used on the same PRIMACS.

PRIMACS-I operates with Prime CPU models P300 and above. PRIMACS-II, requires a Prime P350 or above.

A typical small installation, with a Prime 550, disk drive, printer, one Ontel terminal and Diablo printer, is under $200,000. "But nobody would want it," said Eric W. Knudson, vice-president and director, "because PRIMACS doesn't become competitive until you put eight or nine terminals on the system."

A firm already having a Prime computer can buy the PRIMACS software unbundled. The Ontel terminal for PRIMACS is not standard; it has a special keyboard and three PROMs.

Several PRIMACS systems have already been installed. The Lincoln National Insurance Co. in Fort Wayne, Ind., has 20 terminals now and is "talking about 5000 terminals," according to Knudson. The Bergen Bank in Bergen, Norway, is running PRIMACS on two Prime 550 systems, one in Bergen and the other in Oslo, with 100 terminals and 24 printers.

A PRIMACS system on a Prime 550 has been installed as a part of the automated-office prototype in the Air Force System Command's IMPACT project, or Improved Administrative Capability Test, in Bedford, Mass.

Electronic Message Systems: The Technological, Market and Regulatory Prospects

Kalba Bowen Associates, Inc., Cambridge, Mass.**Massachusetts Inst. of Tech., Cambridge. Center for Policy Alternatives.*Federal Communications Commission, Washington, D.C. Office of Plans and Policy.
AUTHOR: Kalba, Konrad K.; Sirbu, Marvin A. Jr; de Sola Pool, Ithiel; Thompson, Janet Taplin; Clippinger, John H.
E1971A4 Fld: 17B, 45C* GRAI7819
Apr 78 328p*
Contract: FCC-0236
Monitor: FCC/OPP-78/0236
Prepared in cooperation with Massachusetts Inst. of Tech., Cambridge. Center for Policy Alternatives.

Abstract: The report provides an overview of the current status and future prospects of electronic message systems and services (EMS) in the United States. It examines technological and service alternatives, cost and market estimates, and the policy and regulatory issues raised by EMS. Its specific purpose is to assist the Federal Communications Commission in defining a research agenda, which will contribute to the Commission's formulation of EMS-related regulatory policies.

Telecommunication, Trends, Concepts, Substitutes, Data transmission, Computer networks, Market research, Cost estimates, Rates(Costs), Policies, Regulations, Monitoring, Technology assessment, Commerce

Identifiers: *Electronic message systems, Electronic mail, NTISFCC

PB-281 822/7ST NTIS Prices: PC A15/MF A01

Advanced Mail Systems Scanner Technology. Executive Summary and Appendixes A-D

Naval Ocean Systems Center San Diego Calif (393159)

Annual rept. no. 3, 10 Oct 76-9 Oct 77.
E1065J4 Fld: 20F, 46C, 62F, 45E GRAI7812
Oct 77 299p
Rept No: NOSC/TR-170-APP
Monitor: 18

Abstract: The objective of the effort described herein is to provide technical consultation, equipment, and support services to the US Postal Service which will contribute to the development of the system definition of a new-concept processing system, the Electronic Message Service (EMS). Included in the scope of effort are investigations of high-speed image scanning technology, image frame memory storage, image enhancement, and the fabrication of a scanner/frame-store memory test assembly. (Author)

Descriptors: *Optical scanning, *Postal service, *Image processing, Charge coupled devices, Photodiodes, Images, Solid state electronics, Data compression, Image intensification, Automatic control

Identifiers: Electronic message system, Electronic mail, NTISDODXA

AD-A051 508/0ST NTIS Prices: PC A13/MF A01

Framework and Functions of the 'MS' Personal Message System

RAND Corp Santa Monica Calif (296600)

Interim rept.
AUTHOR: Crocker, David H.
E0581E1 Fld: 17B, 9B, 45C, 62A, 62B GRAI7807
Dec 77 72
Rept No: RAND/R-2134-ARPA
Contract: DAHC15-73-C-0181, ARPA Order-189
Monitor: 18

Abstract: This report describes the design, framework, and functions of a system that enables the manipulation of electronic mail, or messages, for users of PDP-11 minicomputers. In the past, electronic mail was available only to users of larger scale computers. The 'MS' Personal Message System transfers the capabilities of electronic mail technology to the Unix operating system, which runs on the DEC PDP-11 hardware. It consists of several pieces of software to compose, transmit, receive, review, and manipulate messages. The system attempts to provide functions which conform to user's pre-existing models of paper based office communications. This report is not a user's manual, but rather a discussion of the functions available, and the issues involved in designing message-processing systems.

Descriptors: *Mail, *Message processing, *Minicomputers, *Communications networks, *Telecommunication, Systems engineering, Computer programs, Message processing

Identifiers: PDP-11 computers, Electronic mail, PMS(Personal Message System), Personal Message System, NTISDODXA

AD-A048 774/4ST NTIS Prices: PC A04/MF A01

Scientific
Time Sharing
Corporation

APL PLUS®

7316 Wisconsin Avenue, Bethesda, Maryland 20014 • (301) 657-8220

*APL*PLUS*

MESSAGE PROCESSING SYSTEM

USER'S GUIDE

February 1975

This manual describes the *APL*PLUS* Message Processing System, a means for rapid intracompany communication to assist users of Scientific Time Sharing Corporation's *APL*PLUS* Service. Methods for sending and receiving messages and using a variety of inquiry programs are included. Steward operations, which control enrollments and allow maintenance of the system, are described in the "*APL*PLUS* Message Processing System Steward's Guide."

Ronald C. Murray (STSC) designed and implemented the *APL*PLUS* Message Processing System. It is based on earlier systems developed by Frank Bates III (Mobility Systems, Inc.) and Lawrence M. Breed (STSC). The present design has profited from the suggestions and experience of many STSC customers and employees since 1971. This User's Guide was composed using the *APL*PLUS* Text Editor and the *APLFULL* Print Train.

*APL*PLUS* MESSAGE PROCESSING SYSTEM

TABLE OF CONTENTS

INTRODUCTION

The *APL*PLUS* Message Processing System, informally called "Mailbox," provides rapid written communication among users of the *APL*PLUS* System, and is intended to assist customers in using the *APL*PLUS* Time Sharing Service more effectively. A Mailbox message may be directed to one person or to many people simultaneously. It may be sent at any time; once sent, it is held in the system until each recipient has received and accepted his copy. A recipient may receive his "mail" at his convenience.

There is no fixed form for messages; they may range from informal one-line notes to carefully edited business or technical documents.

The sender of a message may designate its privacy level and urgency. These designations are solely to alert the recipient, because in fact the privacy of every message is carefully safeguarded.

A message is available to its recipients as soon as it is sent. A recipient can direct incoming messages to his terminal or to an *APL*PLUS* file. Messages directed to a file might be printed later on a high-speed printer at the STSC Computing Center or they might become part of a document storage and retrieval system.

A recipient may forward an incoming message to other people, with optional commentary. The originator's identity and the original distribution list and timestamp (or "postmark") are carried in the forwarded message.

There are programs for receiving messages, sending messages, and making inquiries. The programs are guided, in an interactive fashion, by simple, abbreviated instructions which you enter at your terminal keyboard. All the programs are available in a single "workspace," which in the STSC Mailbox is named *666 BOX*. This particular Mailbox is used primarily between STSC employees and customers. Other workspaces contain similar mailboxes for specific customer organizations and special interest groups.

A person designated as the Mailbox steward is responsible for installing a Mailbox system, enrolling users, defining user groups, and performing other maintenance activities. The steward uses programs in steward workspaces that are companions to the Mailbox workspace. The steward programs are described in the "*APL*PLUS* Message Processing System Steward's Guide."

You will be able to use the Mailbox to send and receive messages once you finish reading this Introduction and learn the use of the three programs named *SEND*, *PRINT*, and *ACCEPT*. The *PRINT* and *ACCEPT* programs are explained on page 6, and *SEND* is described on page 10. A sample terminal session with comments appears on the back cover.

Mailbox terms and phrases

Each Mailbox enrollee has a unique address code of one to five characters (letters and digits). Address codes identify senders and recipients of messages. Each address code is associated with exactly one user; that is, one sign-on number. Generally, an address code is chosen to be a person's initials or a short form of his name. For example, Samuel Clemens might be enrolled as *SLC* or, if that address code were already in use, as *TWAIN*. The *WHOIS* program, described on page 19, will display a person's name, given an address code; or given a name, it will display the address code.

In your company's private Mailbox, employees are usually marked internal and may communicate with anyone who is enrolled. Other enrollees may be marked external and may communicate with those marked internal. They may only communicate with other enrollees marked external under special conditions. In the STSC Company Mailbox, employees are marked internal, while clients, affiliates, and consultants are usually marked external.

A group, which also has a code of one to five characters, is a set of address codes of individuals. A message addressed to a group is sent to each of its members. For example, a group named *MKTG* might hold address codes of those persons interested in correspondence about marketing. When appropriate, an enrollee marked external may be enrolled in one or more groups; in this case, he may communicate with the group as a whole or with any of its members, which may include other enrollees marked external.

A message comprises a message header and some message text. The header displays: the privacy level and urgency; the message number, a unique integer identifying the message; the timestamp, giving the date and time the message was sent; the address code of the sender; and a distribution list. The distribution list shows the address or group codes to which the message was addressed (*TO*) or "carbon copied" (*CC*). Persons who were "blind carbon copied" (*BCC*) also receive the message, although their address or group codes do not appear in the message header.

Here is an example of a message:

NO. 12345 *SENT* 12 *JAN* 1975 11.35.17
FROM MVDR
TO EERO
CC ARCH FLW

TEXTBOOK: I'VE PRINTED THE ROUGH DRAFT OF THE GLOSSARY FROM YOUR FILE. IT LOOKS QUITE GOOD.
PLEASE ADD THIS ITEM.
FRIEZE: THAT PORTION OF THE ENTABLATURE BETWEEN THE ARCHITRAVE AND THE CORNICE, OFTEN ENRICHED WITH SCULPTURE.

The message text is arbitrary character information and may be up to a few thousand characters in length. It is usually entered at a terminal, but it could be obtained from previously stored data. An edited document and formatted results of a computation are examples of stored data which might be used as message text.

A sender may designate the privacy level of his message as *PERSONAL*, *CONFIDENTIAL*, or *NONCONFIDENTIAL*. The meanings of the privacy levels are largely a matter of convention between sender and receiver. *PERSONAL* usually marks a message of a sensitive nature that the sender intends only for the addressed recipients. *CONFIDENTIAL* denotes a somewhat less sensitive message that should be seen only by persons within some agreed-upon group, for instance by company employees.

Clearly, the Mailbox cannot enforce proper use of privacy levels. It does permit a recipient to select for printing only those messages at or below a desired privacy level. Once a message is printed, though, protecting its privacy is the recipient's responsibility.

A sender may also classify a message as *URGENT* and/or *REGISTERED*. Messages marked *URGENT* are printed before any messages not so marked. A message marked *REGISTERED* will cause a confirmation to be returned to the sender whenever the message is received. The *REGISTERED* classification is not made apparent to the message's recipients.

A message sent to you is said to be:

pending — if the message was sent to you, but you have not received it by executing the *PRINT*, *MESSAGE*, *REPEAT*, or *TOFILE* program.

received — if *PRINT*, *MESSAGE*, *REPEAT*, or *TOFILE* has processed the message, but the *ACCEPT* program has not.

accepted — if you have received the message and have executed the *ACCEPT* program.

A message that you have sent is said to be:

active — if some or all of the addressees have not accepted the message. (For them, the message is either in the "pending" or in the "received" state.)

lapsed — if the message has been accepted by all recipients but is still held in the Mailbox file. (Lapsed messages are periodically removed from the Mailbox file, to make room for new messages.)

Prompting

Most keyboard entries are prompted. Before awaiting keyboard input, a Mailbox program will indicate the kind of input wanted. Prompts are normally short; to see a longer prompt giving more detail, respond to a short prompt by pressing RETURN.

The only exception is the entry of message text, which is unprompted. When you are entering message text, pressing RETURN causes an empty line to be added to the text, as you might like to have between paragraphs.

A response of one or more spaces followed by pressing RETURN is represented in this manual and in prompts as SPACE-CR. It selects the last alternative in the prompt and generally has the effect of completing a process, such as ending message text, sending a message, or exiting from a program. On a terminal that has a TAB key, pressing TAB followed by RETURN is equivalent to SPACE-CR.

If a prompt ends with a question mark, you must reply *YES* or *NO* (abbreviated, if you wish, to Y or *N*). SPACE-CR is interpreted as *NO*.

Restarting after an interruption

If program execution is interrupted by an Attention signal, resume execution by entering

```
→GOON
```

(for GO ON), rather than by branching to the suspended line number. The *→GOON* procedure will make sure the Mailbox file is active, reinitialize some internal tables, and resume program execution at the proper restart point (which is often not the line on which the program stopped).

If you interrupt the *PRINT* or *REPEAT* program and then enter *→GOON* , the program asks where you want to restart. You should reply with either a message number or an ordinal number n representing the [n]th message in that session at which to restart the printing.

If you should lose your telephone connection while executing a Mailbox program, sign on again. Your *CONTINUE* workspace is automatically loaded for you. Then, follow one of these two courses. If the terminal first requests input with the carrier at the left margin, just continue what you were entering before the interruption. If, however, the carrier indents six spaces before requesting input, resume execution by entering *→GOON* .

Can't get to a terminal?

If you expect to be away from the Mailbox for a while, there are two special kinds of messages you should consider using. The first is a message addressed to the fictitious enrollee □*NOTE*, which allows you to post a short note like '*I WILL BE ON VACATION AND AWAY FROM A TERMINAL UNTIL* 12/30/74.' Once you have posted a message to □*NOTE*, anyone who includes you in a distribution list will be alerted to your unavailability. (See how to send a □*NOTE* message on page 17.)

In the following example, a sender is advised that *GGG* has posted a □*NOTE* message:

```
      SEND
TO GGG DZ;CC FOX
□NOTE: GGG
DISPLAY? YES
GGG: I AM UNAVAILABLE UNTIL JUNE 1.  DIRECT MESSAGES TO LB.
```

(N.B. In the examples in this manual, the portions entered by the user are shown against a shaded background to distinguish them from those printed by the Mailbox system.)

The second special message is one addressed to □*FILE*. It allows you to have all your messages put into one of your private files, and then have the messages accepted for you. This is important if you receive many messages and will be away for more than a day or two. (□*FILE* is described on page 18.) When you return, if you have a large volume of mail, you can request that it be fileprinted on a high-speed printer at the STSC Computing Center and mailed to you. Programs are available in workspace 1 *FILEPRINT* for high-speed printing of files.

Other Mailbox conventions

- You may force an escape from a Mailbox program by typing:

  O BACKSPACE *U* BACKSPACE *T*

  as the first (and only) entry on a new line. This may lose the entire message you are preparing, but will not lose or alter any other information.

- You may be using a copy of the Mailbox workspace saved in your private library. Such a workspace may become outdated because the original workspace has been superseded by a new version. Your private copy will print a warning report when a newer version of Mailbox becomes available. You may use the *VERSION* program to print the identity and revision level of the copy of the Mailbox you are using (see page 21).

RECEIVING MESSAGES

The *PRINT* and *ACCEPT* programs

The *PRINT* and *ACCEPT* programs are all you normally need for straightforward message reception. In case of problems in printing messages, you may also need the *REPEAT* program. The rest of the programs for receiving messages described in this section are sometimes useful but not essential.

The *PRINT* program acts on pending messages. After *PRINT* has completed, the messages that were printed are considered to be received. When you are satisfied that no messages were lost or garbled (by telephone problems, terminal problems, or running out of paper), executing the *ACCEPT* program will mark the received messages accepted. An example showing the use of *PRINT* and *ACCEPT* appears on the next page and on the back cover.

PRINT

> Prints pending messages, and changes their status from "pending" to "received." Aside from messages marked *URGENT*, which are printed first, messages are printed in message number order.

If no pending messages are marked *PERSONAL* or *CONFIDENTIAL*, printing begins immediately. If any pending messages are *PERSONAL* or *CONFIDENTIAL*, you are prompted with:

> *PERS, CONF, NONCONF:* or *CONF, NONCONF:*

Respond by typing one of these privacy levels in order to print only messages at or below that level. Your response may be abbreviated to *P*, *C*, or *N*. SPACE-CR is interpreted as *NONCONF*. Any other response terminates the *PRINT* procedure without changing the status of any message.

ACCEPT

> Accepts all messages that are in the "received" state, changes the message states from "received" to "accepted," and prints the number of messages accepted.

You should accept your messages as soon as possible after receiving them. Cooperation by all users will result in smoother operation of the Mailbox system. Programs *ACCEPTALLBUT* and *ACCEPTONLY* (see page 9) may also be used to accept mail.

An example showing the person with address code *SRI* printing and accepting some messages follows. *SRI* responds *CONF* to the privacy prompt, so *CONFIDENTIAL* and *NONCONFIDENTIAL* (if any) messages are printed. Message 4027, evidently a response to 4000, was sent by *CLD* who, as a member of the *MATH* group, received message 4000 although he was not explicitly addressed.

```
      PRINT
16 JAN 1975  15.45.00
CONF, NONCONF: CONF
[1] CONFIDENTIAL
NO. 4000  SENT 16 JAN 1975   9.21.53
FROM HARDY
TO   CAYLY SRI SYLV
CC   MATH

ARE THERE ANY INTERESTING NUMBERS BETWEEN 1720 AND 1730?


[2] CONFIDENTIAL
NO. 4027  SENT 16 JAN 1975  13.47.20
FROM CLD
TO   HARDY
CC   CAYLY SRI SYLV

INTERESTING NOS:  HOW ABOUT 1728?  IT'S 12 CUBED.


RECEPTION COMPLETE

      ACCEPT
2 MESSAGES ACCEPTED
```

The *MESSAGE* and *REPEAT* programs

The *MESSAGE* and *REPEAT* programs will work on any message directed to you, whether pending, received, or accepted; and on any message sent by you, whether active or lapsed. However, the Mailbox regularly discards lapsed messages after a time period designated by the steward. Thus if you accept a message sent several days before, it may be discarded almost immediately.

A sender can withdraw a message at any time. (The *WITHDRAW* program is described on page 10; the *WITHDRAW* action is described on page 16.) If you receive a message and the sender later withdraws it, the message will "disappear" from your list of received messages.

r ← *MESSAGE* n

Returns, as result r, a character vector holding the header and text for message number n. The message must have been either sent to you or sent by you. For example, entering *MESSAGE* 3029 prints the header and text of the message with number 3029. Another use of *MESSAGE* is shown here:

```
      EPISTLE←MESSAGE 11533
      EPISTLE↓'FROM'
40  41  3  43
      39↓EPISTLE
FROM PAUL
TO   CORIN

BE NOT DECEIVED; EVIL COMMUNICATIONS
CORRUPT GOOD MANNERS.
```

If you enter an invalid message number for n, the *MESSAGE* program returns an empty vector rather than an error indication.

r ← *MESSAGE* 'abc'

Returns, as result r, a character vector holding the □*NOTE* message for the enrollee with address code abc.

r ← *MESSAGE* '□*NOTE*'
r ← *MESSAGE* '□*FILE*'

Returns, as result r, a character vector holding your □*NOTE* or □*FILE* message, if you have one.

REPEAT n

Prints one or more messages either sent to you or sent by you. n is a vector of message numbers. Any messages previously "pending" are marked "received."

Entering the expression *REPEAT RECEIVED,PENDING* is a convenient way to print all messages that you have not yet accepted. (More details on received and pending messages are on pages 3 and 21.) To print all active messages that you have sent, enter *REPEAT SENT* . (The *SENT* program is described on page 20.)

The *TOFILE*, *ACCEPTALLBUT*, and *ACCEPTONLY* programs

r ← n *TOFILE* file

Copies messages sent to you or by you from the Mailbox to your private file. Any copied messages that were "pending" are marked "received." n holds the message numbers of messages to be copied. If you wish to copy messages to a file that is already tied, file should be the file's tie number, or it can be the file's name. If you wish to store messages in an existing but untied file, file is the name of that file. If you wish to copy messages to a nonexistent file, *TOFILE* will create and tie a file named file.

The result r is a character matrix with as many rows as there are messages. Each row contains the privacy level and urgency, the message number, and the sender's address code.

For example, to copy all pending messages to a file tied with 24, enter the expression *PENDING TOFILE* 24 . To copy both pending and received messages to the file with name *MYMSGFILE*, enter *(RECEIVED,PENDING) TOFILE 'MYMSGFILE'* .

ACCEPTALLBUT n

Like *ACCEPT*, but leaves messages in the "received" state if their message numbers appear in n. For example, if you had ten received messages and were ready to accept all of them except the two with message numbers 4000 and 4027, you may use *RECEIVED* and *ACCEPTALLBUT* like this:

```
      RECEIVED
3955  3998  4000  4003  4012  4027  4028  4035  4042  4075
      ACCEPTALLBUT 4000 4027
8 MESSAGES ACCEPTED
```

ACCEPTONLY n

Like *ACCEPT* and *ACCEPTALLBUT*, but leaves messages in the "received" state if their message numbers do not appear in n. For example, suppose you had used *ACCEPTONLY* in the preceding example. Then:

```
      ACCEPTONLY 4000 4027
2 MESSAGES ACCEPTED
```

You may receive messages from two fictitious enrollees □*RCVD* and □*STEW*. □*RCVD* sends a confirmation whenever someone receives a *REGISTERED* message from you. □*STEW* sends a notice to anyone whose enrollment status in Mailbox is altered by the steward.

SENDING MESSAGES

The *SEND*, *TRANSMIT*, and *WITHDRAW* programs

There are three programs for sending messages: *SEND*, *TRANSMIT*, and *WITHDRAW*.

SEND

Sends a message that you compose at the keyboard using the actions described later in this section.

TRANSMIT x

Sends one message. x is either a variable or an expression containing the message text. If x is a matrix, the message text is formed by appending a carriage return to each row. An example of *TRANSMIT* appears on page 18.

WITHDRAW n

Retracts one or more previously sent messages with message numbers in n from any recipients who have not yet received them. When you withdraw a message, you may either discard the entire message or modify the text, distribution list, or other previous actions and send it again. An example of *WITHDRAW* is on page 18. (*WITHDRAW* may also be used as an action within the *SEND* or *TRANSMIT* program; see page 16.)

Here is the simplest way to send a message. First, execute the *SEND* program by typing *SEND* and then pressing RETURN. When you see the prompt *TO*, respond by typing the address code of the intended recipient (in this example, *EC*) and pressing RETURN.

```
      SEND
TO EC
```

SEND next prompts for message text entry by printing *TEXT* -- unless there is something unusual about the code *EC*. (*EC* might not be an enrolled address code; or might have posted a □*NOTE* message for *SEND* to display before it prompts for text entry.)

```
      SEND
TO EC
TEXT --
THE BASIS OF OUR GOVERNMENT BEING THE OPINION OF THE PEOPLE,
THE VERY FIRST OBJECT SHOULD BE TO KEEP THAT RIGHT; AND WERE
IT LEFT TO ME TO DECIDE WHETHER WE SHOULD HAVE A GOVERNMENT
WITHOUT NEWSPAPERS, OR NEWSPAPERS WITHOUT A GOVERNMENT, I
SHOULD NOT HESITATE A MOMENT TO PREFER THE LATTER.
                                                    /THOMAS
```

Enter the text of your message on as many lines as are needed. To signal the end of the message text, use SPACE-CR. (Simply pressing RETURN at the left margin does not end the message; instead, it adds a blank line to the message text.)

When you have ended a message, the prompt *ACTION:* is printed. Type *SEND*, and the message is sent immediately and its message number and timestamp are printed.

```
ACTION: SEND
NO. 5000  SENT 16 JAN 1787   8.37.05
TO
```

The *TO* prompt indicates that the *SEND* program is ready for you to begin another message. To terminate the process, respond to the *TO* with a SPACE-CR instead of another address or group code.

The preceding example shows the essence of the sending process, using the *SEND* program. The *TRANSMIT* and *WITHDRAW* programs, which provide added flexibility in sending messages, as well as the full facilities of *SEND* are described next.

Details of the message-sending process

The three steps in preparing and sending a message are listed below. The Mailbox system prompt for each is shown in parentheses.

1. (*TO*) Define the distribution list, privacy level, urgency, and any messages to be forwarded along with your message.

2. (*TEXT* --) Start or add to the text of the message.

3. (*ACTION:*) If necessary, add to or correct the entries from steps 1 and 2 and display the revised information; and then either send the message or cancel it.

The *SEND* program goes through each of these steps for every message. Since *TRANSMIT* gets its message text from its right argument, it goes directly from step 1 to step 3. *WITHDRAW* derives both the distribution list and message text from its argument, and so starts with step 3.

The actions performed in steps 1 and 3 are specified in the same way. Following the prompt, you enter one action, or several actions separated by semicolons, selected from the list described below. (The *SEND* program has, in effect, already selected the *TO* action for you in step 1.) To make the first action in step 1 something other than *TO*, either backspace over the *TO* and press ATTN, LINEFEED, or BREAK before typing your action, or follow the *TO* immediately by a semicolon and your action.

At step 1, the program will perform the actions you have typed and then proceed to step 2. At step 3, the program will perform the actions and return to the *ACTION*: prompt until it encounters one of the actions *SEND*, *QUIT*, *CHANGE*, or *MORE*. These actions take the program from step 3 to step 1 or step 2.

You may make mistakes in entering actions; for instance, you may type a nonexistent address code or action name. If you are at step 3, you will receive an error report and the *ACTION*: prompt again. At step 1, you will receive an error report and the prompt "...;". This means you're still at step 1, but the *TO* action isn't being forced on you. In either case, just correct your mistake by retyping the actions that were ignored. For example:

```
      SEND
TO QCE
NOT FOUND: QCE
...;TO QQQ
TEXT --
```

```
      SEND
TO QCE;CC XYZ;CONF
NOT FOUND: QCE
IGNORED ACTIONS: CC XYZ;CONF
...;TO QQQ;CC XYZ;CONF
TEXT --
```

Most of the actions listed below may be performed in step 1 or step 3, but some actions make sense at only one of these points. For instance, the *SEND* action requires that both a distribution list and a message text exist, while the *WITHDRAW* action requires that neither have been entered.

Action names may be abbreviated to the first two characters. *CO;AP* 12345;*SE* is as good as *CONF;APPEND* 12345;*SEND* .

You may include the quote-quad character ⍞ in the actions you type. Before any actions are performed, you will be prompted to enter a character-valued expression to replace the ⍞. Suppose you regularly send messages to the same ten people (who are not enrolled as a group), and you have put their address codes into a character vector named *USUAL*. Then at the *TO* prompt, enter ⍞ . When the ⍞: prompt appears, enter *USUAL* .

Message-marking actions

PERS

Sets the privacy level to *PERSONAL*.

CONF

Sets the privacy level to *CONFIDENTIAL*.

NONCONF

Sets the privacy level to *NONCONFIDENTIAL*. A message is *NONCONFIDENTIAL* unless you have specified *PERSONAL* or *CONFIDENTIAL*.

URG

Marks the message *URGENT*, so that it will be printed before all of the recipients' other messages.

/URG

Removes the *URGENT* marking.

REG

Designates the message *REGISTERED*, so that you will receive a confirmation of its receipt. It is not evident to the recipient whether a message is registered or not.

/REG

Removes the *REGISTERED* designation.

Distribution list actions

The *TO*, *CC*, and *BCC* actions may be used to specify or enlarge the message's distribution list. After *TO*, *CC*, or *BCC*, type one or more individual address or group codes separated by blanks. The codes will be combined with any other codes entered in previous like-named actions.

Each recipient will receive one copy of the message, regardless of the number of times his name may appear in the distribution list. Addressing a message to a group that you are a member of does not send you the message; you must *CC* or *BCC* yourself if you would like to receive a copy.

In the following descriptions, codes represents one or more individual address codes or group codes separated by blanks.

TO codes

Includes these address or group codes in the distribution list and adds them to the *TO* line of the message header.

CC codes

Includes these address or group codes in the distribution list and adds them to the *CC* line of the message header.

BCC codes

Includes these address or group codes in the distribution list but does not include them in the message header.

/TO codes

Removes these address or group codes from the *TO* line of the message. If no codes follow */TO*, then all codes from previous *TO* actions are discarded.

/CC codes

Removes these address or group codes from the *CC* line of the message. If no codes follow */CC*, then all codes from previous *CC* actions are discarded.

/BCC codes

Removes these address or group codes from the *BCC* line of the message. If no codes follow */BCC*, then all codes from previous *BCC* actions are discarded.

Display actions

DIST

Displays the individual address codes from your distribution list. If your message is addressed to any groups, the groups are expanded and the individual members' address codes are displayed instead of the group codes.

PRINT

Displays the message header and text that you have prepared, plus the message numbers of any appended messages. The message header may include a line for the *BCC* list; this line will not be displayed to any recipient.

STATUS

Displays the message header, but not the message text.

Text-altering actions

CHANGE

Lets you make changes to previously entered message text, so that you do not have to type the entire message over again if you need to make any corrections. *CHANGE* is described in the section "Using the *CHANGE* Action," page 22.

EDIT n

Composes the message text into lines of width n. If you omit n, the width is set to 70. You don't need the *EDIT* action unless you're dissatisfied with the widths of the lines you typed in previously. *EDIT* is especially useful after you have made major alterations to the text with *CHANGE*.

MORE

Returns to step 2 of the message-sending process, to add more text to the end of the message. (Use of the *MORE* action is shown on page 16 and in the example on the back cover.) If nearly all remaining space in the workspace is used up, you'll receive the warning report *END MESSAGE SOON*. Stop at a convenient place and send any additional text in a second message.

Message-appending actions

APPEND n

Appends one or more messages, with message numbers in n, to the present message for the purpose of forwarding the appended messages to addressees who were not among the original recipients. The present message may or may not have any text of its own. If the present message has no text of its own, it acts simply as a vehicle for forwarding the received messages. If it does have text of its own, the text serves as commentary concerning the forwarded messages. *APPEND* and its use in forwarding messages is described further on page 17.

/APPEND

Discards all messages collected from previous *APPEND* actions.

Message disposition actions

SEND

Assembles a message based on your distribution list, message text, and any messages being appended; gives the new message a message number and timestamp; and sends it.

QUIT

Discards the distribution list and message text being built, and starts over at step 1.

WITHDRAW n

Retracts one or more messages with numbers in n from any recipients who have not yet received them, and proceeds to step 3 of the message-sending process. When you withdraw a message, you may either discard the entire message (with the *QUIT* action) or modify the text, distribution list, or other previous actions and send the message again. You can withdraw only messages you have sent; that is, you cannot withdraw a message someone else sent. *WITHDRAW* may be an action of the *SEND* or *TRANSMIT* program, only if no distribution list or message text has yet been entered. Here is an example showing the use of the *WITHDRAW* action:

```
      SEND
TO ;WITHDRAW 334
MSG 334 NOT RECEIVED BY MGM
ACTION: MORE
TEXT --
PLEASE LET ME KNOW BEFORE FRIDAY WHICH SCRIPT YOU PREFER.
              (use SPACE-CR to end text)
ACTION: SEND
CONF. NO. 337  SENT  4 JAN 1975  11.16.22
TO
```

Temporary escape action

□

Lets you temporarily exit the sending program in order to enter system commands such as)*MSG*, or execute an inquiry program like *WHOIS*, *RECEIVED*, or *SENT*. If you were using □ to enter system commands, you may enter any number in order to resume. If you were using □ to execute an inquiry program, the sending program automatically resumes with either the *ACTION*: prompt (step 3) or the ...; prompt (step 1). For example:

```
ACTION: □
□:
      )OPR HOW LATE WILL THE SYSTEM BE UP TODAY?/ABC
SENT
OPR:  REGULAR SUNDAY SCHEDULE - UNTIL 17:00.
□:
      3
3
ACTION: CONF
ACTION: □
□:
      WHOIS 'TV'
TV: ABC BBC CBS NBC PBS
ACTION: CC TV;SEND
```

Forwarding messages

The *APPEND* action (page 15) provides a way to forward a message. Suppose you type the action *APPEND* 12345 . When you subsequently type the action *SEND*, two extra things happen. A new line, [*ORIGINATED BY ...*], is added to the header of your message; and message 12345 is placed at the end of your own message text.

A forwarded message is not really part of your message text. In particular, you can't alter its contents by editing your own message. (To be sure, you can send your message, then withdraw it and edit it; but the giveaway line [*ORIGINATED BY ...*] will disappear when you send the message again.)

The privacy level and urgency of your message are whatever you set them to be. They are unaffected by the original privacy or urgency of the message being forwarded.

If you would just like to forward message 12345 to *ABC* without adding any commentary, enter this at step 1:

TO ABC;APPEND 12345;*SEND*

If you would also like the forwarded message to be confidential, type this instead:

TO ABC;APPEND 12345;*CONF;SEND*

Adding some commentary as prefix to the appended message is also easy:

TO ABC
TEXT --
PLEASE REPLY TO THE APPENDED MESSAGE.
(use SPACE-CR to end text)
ACTION: APPEND 12345;*CONF;SEND*

When given a list of message numbers, *APPEND* generates an equal number of messages each having the same commentary as prefix.

Sending a special □*NOTE* message

You may send a message addressed to □*NOTE* (and possibly to other addressees as well) to call some matter to a sender's attention. Anyone who later includes you in a distribution list is invited to see the first line of your □*NOTE* message. He can read the message in full by executing the *MESSAGE* program. The first line of your message should therefore be self-contained and informative; the remainder can give a more complete explanation. (The "first line" extends up to the first carriage return.) An example of a □*NOTE* message follows on the next page.

```
TO □NOTE;CC EAP
TEXT --
OUT OF TOWN OVER THE OCT 31 WEEKEND.
MONTRESSOR HAS INVITED ME TO HIS CHATEAU TO SAMPLE
A CASK OF AMONTILLADO THAT RECENTLY ARRIVED.
(HOW PLEASANT TO HAVE OUR LITTLE DISAGREEMENT
BEHIND US!)
```

When you resume Mailbox activity, you should retract your □*NOTE* message. To do this, enter *WITHDRAW* '□*NOTE*' and type *QUIT* in response to the *ACTION*: prompt. *WITHDRAW* 'abc' , where abc is your own address code, and *WITHDRAW ME* have the same effect.

Sending a special □*FILE* message

While you are out of touch, messages addressed to you may be piling up. To prevent this, put the name of a private file of yours as the first line of the text of a message to □*FILE*. If a file by that name does not exist, the sending program will create one. The program automatically grants the steward access to append and resize the file.

A utility program is run occasionally by the steward to clean out the Mailbox file. Acting on your □*FILE* message, the program will copy all of your pending and received messages to your file (named *MAIL*1223 in the following example) and accept them for you. (Although the steward's program copies the messages, the steward himself cannot see them.)

```
      TRANSMIT 'MAIL1223'
TO □FILE;SEND
MAIL1223 DOES NOT EXIST
CREATE IT? Y
NO. 2321  SENT 23 DEC 1974  16.59.55
```

If you do not wish to have a file created with the name as you specified it, respond *N* to the *CREATE IT?* prompt. The program will respond with the ...; prompt. Reply to this prompt with *CHANGE* if you wish to modify the name, or with *QUIT* to discard the □*FILE* message.

When you resume Mailbox activity, you should retract your □*FILE* message. To do this, enter *WITHDRAW* '□*FILE*' and type *QUIT* in response to the *ACTION*: prompt. For example, the following sequence would retract □*FILE* message 2321.

```
      WITHDRAW '□FILE'
MESSAGE 2321 RECEIVED BY □FILE
ACTION: QUIT
```

MAKING INQUIRIES

Some of the inquiry programs described here, like *WHOIS*, are used mainly for interactive inquiries. Others, like *PENDING*, can be used within a "cover program" which a particular user might write and use instead of one of the standard Mailbox programs.

DIRECTORY

Prints a list of of all individuals whom you may address through the Mailbox, with their address codes and group memberships. The name of an enrollee who is marked external is preceded by an '*e*'.

WHOIS '*abc*'

Prints the address code, name, and group memberships of the person whose address code is *abc*, if there is such a code, and of every person whose last name begins with *abc*. If *abc* is a group code, *WHOIS* prints the address codes of the individual members.

```
      WHOIS 'JQP'
JQP    PUBLIC, JOHN Q.          SALES PROD

      WHOIS 'TV'
TV: ABC BBC CBS NBC PBS
```

To make several inquiries at once, give *WHOIS* a list of items separated by semicolons:

```
      WHOIS 'ISHMAEL;AHAB;GWW'
```

WHOIS '≠*abc*'
WHOIS *n*

Prints the information described above for *abc* or sign-on number *n*, plus some additional enrollment information. *WHOIS* can be used in this way only by enrollees who are marked internal.

PREVIEW

Tells how many messages you have pending, who sent them, and whether any are marked *URGENT*. For example:

```
      PREVIEW
3 MESSAGES PENDING FROM ABC BBC CBS
1 URGENT MESSAGE
```

r ← *SENT*

Returns information on messages you have sent, including the message number and individual recipients' address codes for each active message. The codes *P*, *C*, *U*, or *R* represent *PERSONAL*, *CONFIDENTIAL*, *URGENT*, or *REGISTERED* messages. An asterisk appears next to the address code of each person who has received the message but not accepted it. Recipients who have accepted the message are not shown. An example of using *SENT*:

```
      SENT
  C     786 *CBS
 RP     801 *PBS
U C     805  NBC    *CBS
        829 *ABC     PBS    CBS
  C     850 *ABC     NBC
U       853  NBC
```

UNREAD

Equivalent to executing *PREVIEW* and then *SENT*.

r ← *INDEX* 'x'

Returns information about the type of message selected by the argument 'x'. The resulting matrix r includes the urgency, privacy level, message number, and sender's address code. x may be any one of the following characters:

L Requests information on lapsed messages that you have sent.

P Requests information on pending messages.

R Requests information on received messages.

A Requests information on active messages that you have sent.

For example:

```
      INDEX 'P'
U C     816 CBS
  C     831 ABC
  P     845 BBC
```

r ← *PENDING*

Returns, as result r, the numbers of all pending messages. For example:

```
      PENDING
816   831   845
```

r ← *RECEIVED*

Returns, as result r, the numbers of all messages you have received but not yet accepted. For example:

```
      RECEIVED
777   798   803   807   810
```

r ← *LAPSED*

Returns, as result r, the numbers of those messages sent by you that have been accepted by all recipients but not yet discarded from the Mailbox.

r ← *ME*

Returns, as result r, your address code as a character vector.

VERSION

Prints the identity and revision level of the copy of the Mailbox you are using.

ACCEPTED

ACCEPTED is a variable which is set by the *ACCEPT*, *ACCEPTALLBUT*, and *ACCEPTONLY* programs. It contains the message numbers of the messages you have accepted. Unless you save a private copy of the Mailbox, *ACCEPTED* is lost when you load another workspace. *ACCEPTED* is assigned a new value each time you execute *ACCEPT*, *ACCEPTALLBUT*, or *ACCEPTONLY*.

USING THE *CHANGE* ACTION

General procedure

CHANGE permits detailed editing of previously entered message text. It allows you to make corrections, insertions, and deletions to the text so that you do not have to type the entire message over again.

When you enter *CHANGE* (or *CH*) in response to the *ACTION:* prompt, the first 35 characters of the text are printed at the terminal. You can then delete or replace any of these characters by typing control characters under the appropriate text characters. There are also control characters for inserting new characters in the text. If you enter *CHANGE* before you have supplied any text, *CHANGE* automatically starts you off in text entry mode.

At any time, you can move the "window," the 35 characters through which the text is viewed, to any other part of the text that you wish to alter. Use of *CHANGE* requires knowledge of the control characters for deleting and inserting text and moving the window to different parts of the message text. To escape from the *CHANGE* action, use SPACE-CR.

The *CHANGE* control characters are described in this section. In each description, the control character is printed at the left margin exactly like the character used at the terminal. In addition, the following notation is used:

n — a number representing a count of characters in the text, as entered by the user.

string — a sequence of characters entered by the user.

An example showing the use of *CHANGE* appears on page 28.

Controlling the window

When *CHANGE* begins, the window is at the beginning of the text. To make changes elsewhere in the text, move the window by typing any one of the following control characters:

F

Moves the window forward to display the text segment immediately following the segment currently displayed.

Fn

Moves the window forward n characters from the current position (or to the end of the text if n is larger than the number of characters remaining in the text).

B

Moves the window backward to display the text segment immediately preceding the segment currently displayed.

*B*n

Moves the window backward n characters from the current position (or to the start of the text if n is larger than the number of characters available).

n

Moves the left position of the window to the nth character position in the text. For example, to return the window to the beginning of the text at any time, enter the number 0.

*L*string

Locates the first occurrence of string in the text following the character position under which the *L* is typed, and moves the window forward to that text segment; or it prints the report *NOT FOUND...*, if appropriate, without moving the window. After the first move, entering *L* without string continues the search forward for the next occurrence of string. The *L*, *K*, and *E* control characters share a "common memory;" thus, if entering *LWUNGA* doesn't find *WUNGA* forward, type *K* to search backward.

*K*string

Locates the most recent occurrence of string in the text preceding the character position under which the *K* is typed, and moves the window backward to that text segment; or it prints the report *NOT FOUND...*, if appropriate, without moving the window. After the first move, entering *K* without string continues the search backward for the next occurrence of string.

*E*string

Locates the earliest occurrence of string in the text, and moves the window forward or backward to that text segment; or it prints the report *NOT FOUND...*, if appropriate, without moving the window. After the first move, entering *E* without string continues the search.

*W*n

Sets the width of the window to n characters. The maximum window size is 100 characters; the default is 35.

Making simple deletions and insertions

.string

Inserts character string immediately preceding the character above the period.

//// or \\\\

Deletes the characters above the slashes. Any number of slashes may be used but they must be contiguous.

////string or \\\\string

Deletes the characters above the slashes and replaces them with the string of characters that follows. Any number of slashes may be used but they must be contiguous. (N.B. If the replacement string is to begin with a slash (/), use the backslash (\) for the control character.)

Changing large segments of text

*R*string

Replaces every occurrence of string, starting at the position above the *R* and proceeding to the end of the text, by another string of characters. First, enter *R*string. If string is not found, the report *NOT FOUND...* is printed, and the current window is displayed again.

If string is found, *CHANGE* prompts with *REPLACE BY* --. Respond by entering the replacement string. If you made a mistake in typing string, you may recover at this point by retyping exactly what you first typed for string. If you respond to *REPLACE BY* -- by pressing RETURN (with no replacement string), every occurrence of string is deleted.

After the replacements are made, the number of replacements is printed, followed by the prompt *DISPLAY*?. If you respond *YES* (or Y), all replacements are printed (in context) so that you may correct any unanticipated replacements.

*S*string

Same as *R*string except that *S*string replaces every occurrence of string from the beginning of the text up to but not including the position under which the *S* is typed.

*D*string

Deletes characters from the text, starting at the position above the *D* and ending immediately after the next occurrence of string. If the requested deletion involves 200 or more characters, *CHANGE* asks *DELETE n CHARACTERS?*. Reply *Y* or *N*.

*M*string

Moves the portion of text starting at the position above the *M* and ending with (and including) the next occurrence of string to a location forward in the text. First, enter *M*string. If string is not found, the report *NOT FOUND...* is printed, and the current window is displayed again.

If string is found, *CHANGE* prompts with *TO FOLLOW* --. Respond by entering another string of characters. The text to be moved will be relocated just after this string. If you decide at this point that you would like to abort the move, respond to the prompt *TO FOLLOW* -- by just pressing RETURN. The report *NOT FOUND...* is printed, and the current window is displayed again.

*N*string

Same as *M*string except that *N*string moves the designated portion of text to a location backward in the text. *N*string places the text to be moved immediately before the string that you enter in reply to the prompt *TO PRECEDE* --.

*C*string

Same as *M*string except that *C*string copies the portion of text rather than moving it. The original portion is not deleted. A copy is only performed in a forward direction.

Adding more text

I or *I*string

Prepares to insert more text in the message immediately preceding the character above the *I*. You may begin typing the insert on the same line immediately following the *I*. After you press RETURN, *CHANGE* prints a number which represents the last character position before the insert. You may continue typing additional lines to be inserted. When you have finished entering the text to be inserted, either use SPACE-CR to return to normal *CHANGE* mode which will then accept other control characters, or enter another SPACE-CR to terminate the *CHANGE* program.

Z or *Z*string

Same as *I*string except that *Z*string prepares to add one or more lines of text to the end of the message. After you type *Z* or *Z*string and press RETURN, *CHANGE* prints a number which represents the last character position in the text before the addition.

J

Prepares to insert an *APL* character-valued expression immediately before the character above the *J*. After you enter *J*, the prompt □: is printed. At this point, you may enter a character vector or matrix, or any *APL* expression that returns a character vector or matrix.

When you're using *I*, *Z*, or *J*, the warning report *END MESSAGE SOON* will be given if nearly all remaining space in the workspace is used up. If you receive this report, stop at a convenient place and send any additional text in a second message.

Displaying text

*P*string

Prints the portion of text starting at the position above the *P* and ending after the next occurrence of string. Entering *P* without string displays the entire text starting at the position above the *P* and proceeding to the end of the text.

*V*string

Prints a window containing a view of each occurrence of string, starting at the present window position and proceeding to the end of the text.

*U*string

Same as *V*string except that *U*string gives a view of each occurrence of string from the beginning of the text up to but not including the position above the *U*.

Inserting nontypable characters

Certain typewriter operations (backspace, carriage return, idle, linefeed, and null) cannot be entered as individual characters directly from a terminal keyboard. Therefore, these operations require special representation to permit editing within the text. The following character combinations are used to represent these operations:

Backspace	'¢B'
Carriage return	'¢C'
Idle	'¢I'
Linefeed	'¢L'
Null	'¢N'

During original text entry you need not be concerned with these combinations. In fact, they can only be inserted using *CHANGE*. However, when text is displayed during *CHANGE*, these combinations are shown instead of the related typewriter operations. One of these combinations may be entered at any text modification point to stand for the corresponding typewriter operation. When the message is printed, the appropriate typewriter operation will occur.

By using *EDIT* after you have used *CHANGE*, you need not normally be concerned with the location of carriage returns, because *EDIT* will align right and left margins for any width desired.

Example

An example showing the use of the *CHANGE* action appears on the next page. What the user types and the program responses are shown on the left side. Explanatory notes appear on the right.

```
TO EDTR
TEXT --
OUR STOCK OF APL*LUS FINANCEPAK II GUIDES
IS RUNNING LOW.  WE HAVE ONLY 10 COPIES LEFT.  SHALL
WE HAVE THE CURRENT VERSION REPRINTERED.

ACTION: CH
OUR STOCK OF APL*LUS FINANCEPAK II    35
.RE: FINANCEPAK II USER'S GUIDES
UIDESOUR STOCK OF APL*LUS FINANCEPA   61
                        .P
APL*PLUS FINANCEPAK II GUIDES<CIS R   79
LUIDESOUR
NOT FOUND...
APL*PLUS FINANCEPAK II GUIDES<CIS R   79
K
UIDESOUR STOCK OF APL*PLUS FINANCEP   61
     .<C
DES<COUR STOCK OF APL*PLUS FINANCEP   63
F
AK II GUIDES<CIS RUNNING LOW.  WE H   98
     . USER'S
SER'S GUIDES<CIS RUNNING LOW.  WE H  105
L10
10 COPIES LEFT.  SHALL<CWE HAVE THE  149
/7
NLY 70 COPIES LEFT.  SHALL<CWE HAVE  145
LREPRIN
REPRINTERED.<C  180 [END]
       ////D?
NTED?<C  178 [END]

ACTION: PRINT
TO    EDTR

RE: FINANCEPAK II USER'S GUIDES
OUR STOCK OF APL*PLUS FINANCEPAK II USER'S GUIDES
IS RUNNING LOW.  WE HAVE ONLY 70 COPIES LEFT.  SHALL
WE HAVE THE CURRENT VERSION REPRINTED?

ACTION: CH
VE THE CURRENT VERSION REPRINTED?  178 [END]
Z
178
IF WE DO DECIDE TO REPRINT, HOW MANY COPIES DO YOU
THINK WE NEED?

NEED?  244 [END]

ACTION: CONF;SEND
```

```
Specify main recipient of this message.
Prompt for text entry.
Message
   text
      entry.
Use SPACE-CR to end text entry.
Type CH to invoke CHANGE action.
First window of text displayed.
Use . control character to insert words before "OUR".
Window near end of insertion displayed.
Insert "P" before "L".
Window displayed.
Locate "UIDESOUR" to insert carriage return.
"UIDESOUR" was not found forward in the text.

Type K to look backward in the text for "UIDESOUR".

Insert special characters for a carriage return.

Type F to move the window forward one window width.

Insert "USER'S"; note the extra space before.

Locate "10" in the text.
Text segment starting with "10" displayed.
Delete "1" and replace it with a "7".

Locate "REPRIN".

Correct "REPRINTED" spelling and change "." to "?".

Use SPACE-CR to exit the CHANGE action.
Have message printed to see if it's O.K. now.

Would like to add a few words to text.
Last window of text displayed; character count.
Type Z to add more text.
Last character position before addition.
Enter as many additional lines as you like.

Use SPACE-CR to end text entry.
Last window displayed.

Mark message CONFIDENTIAL and send it.
```

INTERPRETING REPORTS FROM THE MAILBOX SYSTEM

During a Mailbox session, reports which inform you that you have made an error or which just provide you with some information are printed at the terminal. Possible reports from the Mailbox system are listed here, together with the names of the programs where they may occur, the cause of the report, and a suggested remedy. When the report is mainly informational, no remedy is given.

ACCEPT RECEIVED MESSAGE(S)

Program: *PREVIEW*, *SEND*, *TRANSMIT*, *UNREAD*, or *WITHDRAW*
Cause: You have received but not accepted one or more messages that were sent to you.
Remedy: Use the *ACCEPT*, *ACCEPTALLBUT*, or *ACCEPTONLY* program to accept your received messages.

ADDRESS CODES OR SPACE-CR:

Program: *SEND* or *TRANSMIT*
Cause: You pressed RETURN immediately following the *TO* prompt, without first supplying any address codes.
Remedy: If you wish to send a message, type address or group codes for the *TO* distribution list. If you wish to leave the *SEND* program, use SPACE-CR.

APPENDING n *MESSAGE(S) TO THE FILE* fileid

Program: *TOFILE*
Cause: The *TOFILE* program prints the current date and time followed by this report of the total number n of messages being written to file fileid.

"APPEND' MUST BE FOLLOWED BY ONE OR MORE MESSAGE NUMBERS

Program: *SEND*, *TRANSMIT*, or *WITHDRAW*
Cause: You have attempted to use the *APPEND* action with no message numbers.
Remedy: Reenter the action *APPEND* followed by one or more message numbers.

BYPASSING URGENT MESSAGE(S) FROM codes

Program: *PRINT*
Cause: You have one or more pending messages marked *URGENT* by their senders, but you have specified that only messages with a certain privacy level be printed. The urgent messages having a higher level of privacy than specified are bypassed.
Remedy: Repeat the *PRINT* program request, specifying a higher level of privacy.

CANNOT APPEND TO □FILE

Program: *SEND*, *TRANSMIT*, or *WITHDRAW*
Cause: You have attempted to append a message using the *APPEND* action to a message addressed to *□FILE*.
Remedy: Use the *QUIT* action to discard the *□FILE* message, or use the *SEND* action to send it.

file *DOES NOT EXIST*
CREATE IT?

Program: *SEND*, *TRANSMIT*, or *WITHDRAW*
Cause: You have attempted to send a *□FILE* message but have supplied the name file for a file that does not exist in your library.
Remedy: If you wish the program to create a file with this name, respond Y for *YES*; if not, respond *N* for *NO*.

END MESSAGE SOON

Program: *SEND*, *TRANSMIT*, or *WITHDRAW*
Cause: Nearly all remaining space in the workspace has been used up.
Remedy: Stop entering text at a convenient place and send additional text in a second message.

EXT: codes

Program: *SEND*, *TRANSMIT*, or *WITHDRAW*
Cause: You have included in the distribution list one or more address codes of enrollees who are marked external. These persons are probably not employees of your company.
Remedy: Alter the distribution list by using */TO*, */CC*, or */BCC* if you do not want any of these people to receive the message.

=== □*FILE MESSAGE ACTIVE* ===

Program: The first Mailbox program that you execute in a session.
Cause: You have a □*FILE* message active.
Remedy: Ignore the report if the □*FILE* message is still valid, or use the *WITHDRAW* program and the *QUIT* action to discard it.

□*FILE MESSAGES CAN ONLY BE SENT ON YOUR HOME (THE OTHER) SYSTEM*

Program: *SEND*, *TRANSMIT*, or *WITHDRAW*
Cause: You have attempted to send a □*FILE* message on your "non-home" system.
Remedy: Use the *QUIT* action to discard the message. Then, prepare and send a □*FILE* message on your "home" system.

IGNORED ACTIONS: z

Program: *SEND*, *TRANSMIT*, or *WITHDRAW*
Cause: As a result of the error described in the report on the preceding line, the actions z could not be processed.
Remedy: Reenter the ignored actions.

MAILBOX SYSTEM ERROR.
PLEASE REPORT THE CIRCUMSTANCES THAT LED TO THIS PROBLEM TO THE STSC COMPUTING CENTER.

Program: Any Mailbox program.
Cause: There is an error in a Mailbox program.
Remedy:)*SAVE* the workspace and keep the terminal printout of this session. Then report the problem to the *APL* operator at the STSC Computing Center by sending a)*OPR* message.

MESSAGE n *NOT FOUND*

Program: *REPEAT*, *SEND* (*APPEND* or *WITHDRAW* action), or *WITHDRAW*
Cause: You have specified a message number n for a message that does not exist. If you were using arguments like *PENDING* or *RECEIVED*, this report indicates that the message has been withdrawn by its sender.
Remedy: Reenter the action using a valid message number.

n *MESSAGE(S) ACCEPTED*

Program: *ACCEPT*, *ACCEPTALLBUT*, or *ACCEPTONLY*
Cause: You have just accepted n received messages.

MESSAGES ADDRESSED TO □FILE MUST NOT BE ADDRESSED TO ANYONE ELSE

Program: *SEND*, *TRANSMIT*, or *WITHDRAW*
Cause: You have addressed a message to *□FILE* but have also included other address or group codes in the distribution list.
Remedy: Use */TO*, */CC*, or */BCC* to alter the distribution list.

MESSAGES MAY NOT BE SENT TO □STEW OR □RCVD

Program: *SEND*, *TRANSMIT*, or *WITHDRAW*
Cause: You have included *□STEW* or *□RCVD* in the distribution list of the message.
Remedy: Use */TO*, */CC*, or */BCC* to remove *□STEW* or *□RCVD* from the distribution list.

n *MESSAGE(S) PENDING*

Program: *SEND*, *TRANSMIT*, or *WITHDRAW*
Cause: You have not yet received one or more messages that were sent to you.
Remedy: Use the *PRINT*, *MESSAGE*, *REPEAT*, or *TOFILE* program to receive your pending messages.

n *MESSAGES PENDING FROM* codes

Program: *PREVIEW* or *UNREAD*
Cause: You have not yet received one or more messages sent to you by persons with these address codes.
Remedy: Use the *PRINT*, *MESSAGE*, *REPEAT*, or *TOFILE* program to receive your pending messages.

MSG n *NOT RECEIVED BY* codes

Program: *WITHDRAW*, or the *WITHDRAW* action within the *SEND* program.
Cause: The persons with these address codes have not yet received the message with message number n.

MSG n *RECEIVED OR TRANSFERRED:* codes
NOT RECEIVED BY codes

Program: *WITHDRAW*, or the *WITHDRAW* action within the *SEND* program.
Cause: The persons with these address codes have already received the message with message number n, or the message has been transferred to the other system. Persons who have not yet received the message are identified on the second line of the report.

MUST BE A CHARACTER VALUE
TRY AGAIN...

Program: *SEND*, *TRANSMIT*, or *WITHDRAW*
Cause: You have used ? following a *TO* or *ACTION:* prompt, but the evaluated input you supplied was numeric instead of character-valued.
Remedy: Reenter the input correctly. Entering a space satisfies the ?.

NO DISTRIBUTION LIST GIVEN

Program: *SEND*, *TRANSMIT*, or *WITHDRAW*
Cause: You have attempted to send a message but have not specified any recipients of the message. Possible reasons are:
- You pressed Attention before receiving the *TO* prompt.
- You typed a semicolon immediately following the *TO* prompt.
- You removed all addressees by using the */TO*, */CC*, or */BCC* actions.
- All address codes were invalid.

Remedy: Use the *TO*, *CC*, or *BCC* action to specify intended recipients of the message in a new distribution list.

NO MAIL FOR abc

Program: *PRINT*
Cause: There are no messages pending for the enrollee with address code abc.

privacy *NO.* n *SENT* date time

Program: *SEND*, *TRANSMIT*, or *WITHDRAW*
Cause: You have just sent a message. This report indicates the message number n and the date and time the message was filed in the Mailbox system. If you marked the message *CONFIDENTIAL* or *PERSONAL*, that privacy level appears in the report as '*CONF*'. or '*PERS.*'.

NOT AN ACTIVE MESSAGE!

Program: *REPEAT*
Cause: In the argument to *REPEAT*, you have included the message number of a message that has been withdrawn by its sender.
Remedy: Execute another Mailbox program, if you wish.

NOT A TIE NUMBER: tn

Program: *TOFILE*
Cause: As the right argument to *TOFILE*, you have given a number tn that is not the tie number of any file.
Remedy: Reenter the *TOFILE* statement, using a file name or a valid tie number as the right argument.

NOT A VALID ACTION: z

Program: *SEND*, *TRANSMIT*, or *WITHDRAW*
Cause: You have followed an *ACTION:* prompt with something unrecognizable as a valid action. For example, you may have attempted to specify a distribution list without preceding it with *TO*, *CC*, or *BCC*.
Remedy: Reenter the actions correctly.

NOT A VALID FILE NAME

Program: *SEND*, *TRANSMIT*, or *WITHDRAW*
Cause: The file name that you have specified in the text of a message to □*FILE* does not conform to the naming conventions for *APL*PLUS* files.
Remedy: Use the *QUIT* action to discard the message, or use the *CHANGE* action to modify the message text. Refer to the *APL*PLUS* File Subsystem Instruction Manual (STSC, 1971) for a complete description of file names.

=== □*NOTE MESSAGE ACTIVE* ===

Program: The first Mailbox program that you execute in a session.
Cause: You have sent a □*NOTE* message, which is still active.
Remedy: Ignore the report if the □*NOTE* message is still valid, or use the *WITHDRAW* program and the *QUIT* action to discard it.

□*NOTE(S)*: codes
DISPLAY?
FROM

Program: *SEND*, *TRANSMIT*, or *WITHDRAW*
Cause: You have included in a distribution list one or more addressees who have posted □*NOTE* messages.
Remedy: If you would like to have the first line of any of the □*NOTE* messages printed, reply *Y* to the *DISPLAY?* prompt; otherwise, reply *N* or SPACE-CR. When you receive the *FROM* prompt, type the address codes of those persons whose □*NOTE* messages you would like to see. If there is only one □*NOTE* message, the *FROM* prompt is not given.

NOT FOUND...

Program: *SEND*, *TRANSMIT*, or *WITHDRAW*
Cause: The string of characters you have typed following the *K*, *L*, *E*, *R*, *M*, *N*, or *C* control character in *CHANGE* cannot be located, or your response to the *TO FOLLOW* -- or the *TO PRECEDE* -- prompt was invalid.
Remedy: Determine your mistake and enter the statement correctly.

NOT FOUND: codes

Program: *SEND*, *TRANSMIT*, or *WITHDRAW*
Cause: You have entered one or more invalid address or group codes in the *TO*, *CC*, *BCC*, */TO*, */CC*, or */BCC* actions.
Remedy: Reenter the action with valid codes. If you do not know a person's address code, enter □ in response to the *ACTION*: prompt and then use the *WHOIS* program to find out the code (see page 16).

NOT WITHIN A SUSPENDED FUNCTION

Program: Any Mailbox program.
Cause: You have attempted to execute a Mailbox program while another program is suspended.
Remedy: Enter →*GOON* to continue with the suspended program, or just enter → to terminate the program immediately.

NOT YOUR MESSAGE: n

Program: *TOFILE*
Cause: In the left argument to *TOFILE*, you have specified the message number of a message that was not sent either to you or by you.
Remedy: The message with message number n is not appended to the file; all others are.

NO VALID TEXT GIVEN

Program: *SEND*, *TRANSMIT*, or *WITHDRAW*
Cause: You have attempted to send a message without supplying any message text.
Remedy: Use the *MORE* or *CHANGE* action to enter some text; or use the *APPEND* action to append (forward) another message; or use the *QUIT* action to discard the message.

ONLY SINGLE MESSAGE NUMBERS ALLOWED

Program: *MESSAGE*
Cause: You have specified more than one message number in the argument to the *MESSAGE* program.
Remedy: Reenter the statement using only one message number, or use the *REPEAT* program to print more than one message.

[*ORIGINATED BY* abc *NO.* n *SENT* date time]

Program: *PRINT*, *REPEAT*, or *MESSAGE*
Cause: The message with message number n, which was originally sent by the person with address code abc, was forwarded to you in the message currently being printed.

RECEPTION COMPLETE

Program: *PRINT*
Cause: All pending messages at or below the privacy level you specified have been printed.
Remedy: If you have more pending messages with a higher privacy level, use the *PRINT* program again to print those.

REG. privacy *NO.* n *SENT* date time

Program: *SEND*, *TRANSMIT*, or *WITHDRAW*
Cause: You have just sent a message marked *REGISTERED*. This report indicates the message number n and the date and time the message was filed in the Mailbox system. If you marked the message *CONFIDENTIAL* or *PERSONAL*, that privacy level appears in the report as '*CONF.*' or '*PERS.*'.

RESTART AT MSG n

Program: →*GOON*
Cause: The *PRINT* or *REPEAT* program was interrupted, and you entered →*GOON* to restart the program.
Remedy: The Mailbox restart mechanism requires that you enter either a message number n, or an ordinal number n so that the program resumes printing at the [n]th message. If you wish to stop the program, type *O* BACKSPACE *U* BACKSPACE *T* to escape.

SOME OF YOUR MAIL IS MARKED privacy.
WHICH TYPES OF MAIL DO YOU WISH TO PRINT?

Program: *PRINT*
Cause: You pressed RETURN in response to the privacy prompt *CONF, NONCONF:* or *PERS, CONF, NONCONF:*.
Remedy: Respond *N* to print nonconfidential messages, *C* to print confidential and nonconfidential messages, or *P* to print all pending messages.

TRY AGAIN...

Program: *SEND*, *TRANSMIT*, or *WITHDRAW*
Cause: You have typed some character that *CHANGE* does not recognize as a valid control character.
Remedy: Reenter the command correctly.

n *URGENT MESSAGE(S)*

Program: *PREVIEW* or *UNREAD*
Cause: You have n pending messages that are marked *URGENT*.
Remedy: Use *PRINT*, *REPEAT*, *MESSAGE*, or *TOFILE* to receive these messages as soon as possible.

URGENT privacy *NO.* n *SENT* date time

Program: *SEND*, *TRANSMIT*, or *WITHDRAW*
Cause: You have just sent a message marked *URGENT*. This report indicates the message number n and the date and time the message was filed in the Mailbox system. If you marked the message *CONFIDENTIAL* or *PERSONAL*, that privacy level appears in the report as '*CONF.*' or '*PERS.*'.

URGENT REG. privacy *NO.* n *SENT* date time

Program: *SEND*, *TRANSMIT*, or *WITHDRAW*
Cause: Same as the preceding report except that this message was also marked *REGISTERED*.

VALUE ERROR

Program: An *APL* error.
Cause: You have mistyped the name of a Mailbox program.
Remedy: Reenter the statement correctly.

WITHDRAW IS NOT A VALID ACTION HERE. DISPOSE OF PRESENT MESSAGE.

Program: *WITHDRAW*
Cause: You have attempted to use the *WITHDRAW* action within the *WITHDRAW* program.
Remedy: Use the *QUIT* action to discard the present message, or use the *SEND* action to send it.

WS FULL

Program: An *APL* error.
Cause: You have used up all remaining space in the Mailbox workspace.
Remedy: If you are using a private version of the Mailbox workspace to which you have added functions, try using the parent Mailbox workspace. Otherwise, contact your Mailbox steward or the *APL* operator for assistance.

WS NOT FOUND

Program: An *APL* error.
Cause: You have made an error in typing the name or library number of the Mailbox workspace.
Remedy: Reenter the)*LOAD* statement correctly.

YOU ARE NOT ENROLLED IN THE STSC MAILBOX SYSTEM.
PLEASE SEE YOUR STSC MARKETING REPRESENTATIVE FOR DETAILS
IF YOU WISH TO BE ENROLLED.

Program: The first Mailbox program that you execute in a session.
Cause: You have attempted to use a Mailbox system in which you are not enrolled.
Remedy: Follow the given instructions.

APPENDIX A

Use and Misuse of the Mailbox

This section describes STSC's recommended use of the Mailbox for its employees, and was adapted from a paper written by Philip S. Abrams (STSC).

Mailbox is a powerful and sophisticated communication tool. As with any tool, it is not sufficient to know the purely technical aspects of its use. Mailbox is also a social phenomenon, and there is a body of experience and convention that has accumulated over the period it has been in use. These notes are in the form of questions to consider when using Mailbox.

Is this message necessary?

Many people receive large numbers of messages daily, both from customers and from other internal users. Every message takes time to print, to read, and to answer. Questioning the necessity of a message before sending it can save time for both the sender and the recipients. Do not, of course, use this as an excuse not to communicate. A short message that you've taken some action can allow the recipient to stop thinking about the need for action and go on to other matters. Simply be aware of the aggregate time consumed by all of the people involved in a message. A good practice is to clock the print time, multiply the result by the number of recipients, and judge whether the topic justifies the total terminal-hours required to print the message.

Is Mailbox the best way to communicate this information?

Mailbox is habit-forming, and it is easy to forget that other means of communications exist. Some subjects are better treated by telephone, through the postal mail, or in a face-to-face meeting.

Is the distribution appropriate?

Are you sending this message to the right people? Are they all really interested? Have you omitted anyone? Is the distribution list as short as possible?

Is the message too long?

A long message is an imposition on the recipient. Obviously there is no absolute gauge of length. The length of a message relates to the content and importance of the message, the addressees, and the expected response.

If a long message is really necessary, there are other possible ways to communicate it. One technique is to put the edited text into a file or workspace and then to send a short Mailbox message telling people where to find the text. Also, long messages that are less timely might better be sent through the mails.

What impression does this message give of the sender?

Written communication exposes the sender in a different way than verbal communication. In particular, spelling, language, and grammar stand out, as well as typing skill. Although not everybody is a Hemingway or Shakespeare, the form of a message influences how the recipient feels about it and about the sender.

The *PRINT* and *CHANGE* actions in the *SEND* program are intended to allow you to read and revise a message before sending it. *QUIT* permits you to change your mind before sending, while *WITHDRAW* lets you do so later.

Remember also that people other than those on the original distribution list may eventually see a message, through forwarding procedures and copying machines.

Some messages sound awkward or antagonistic when put into writing. If you feel this upon rereading your message (before sending), maybe a phone call or a meeting would be a better way to communicate.

In the same vein, be aware of this "limited bandwidth" of the Mailbox when receiving a message. Often an apparently curt or hostile-sounding message was not intended that way. Mailbox is marvelous for transmitting objective information, but can break down when feelings are involved. If in doubt, do not hesitate to clear up misunderstandings with further discussion through Mailbox, by telephone, or in person.

Is this message overclassified? Underclassified?

The following guidelines are helpful:

PERSONAL: For your eyes only. Print only in private, and treat such mail as very private and personal.

CONFIDENTIAL: For company internal distribution only. Generally, this classification is used for those messages that you would not want a competitor or a customer to see.

NONCONFIDENTIAL (unclassed): Everything else.

URGENT: A message of any privacy level can be marked *URGENT*. These messages will be printed before any others. If a subject is extremely urgent, don't rely on Mailbox to assure its receipt; telephone instead.

REGISTERED: Any message may also receive this marking. The time and date of the recipient's reception will then be reported. Remember though, that receipt of a message does not always mean it has been read.

Is the message pertinent?

Each message should treat one subject at a time. A handy convention is to summarize the topic in a few words at the beginning of a message. This allows recipients quickly to identify messages related to a given topic. Use separate messages for separate subjects.

Have I properly disposed of all messages?

Some messages are informational and need no reply. Others need answers. If you cannot answer questions posed in a message, tell the sender so. If there will be a delay in your answer, tell the sender about when to expect the answer. As the number of Mailbox enrollees grows, messages tend to get misdirected. If you know someone who can answer a question, forward the message to that person, and if appropriate, tell the sender you have done so.

Replies should be timely. It is easy to create misunderstandings when a direct message is not answered. The sender does not know why he has gotten no answer. The problem is not as serious when you receive a message as a member of a group. In general, it is not necessary to reply to "broadcast" messages if you cannot help.

In replying to a broadcast message, it is usually not necessary to send copies to all original recipients. However, since people may be interested in your reply, it may be helpful to keep a copy of the return message, in a

file or workspace. Be sure to advise the group of the location of the return message.

Who else should see this message?

Messages are sometimes not sent to all interested or concerned parties. Use *APPEND* to remedy this. Do not hoard information.

Miscellaneous suggestions for Mailbox use

If you receive a message that you feel is a waste of your time, tell the sender. He may not realize that you are not interested in the topic.

Your mail is private - so is other peoples'. Never look at somebody else's mail either while it is printing or when it is lying around. If a message was meant for you, you would have received it. A simple rule is to treat others' mail as if it were in a perfumed pastel envelope.

Your sphere of communication in the Mailbox is usually larger than your manager, subordinates, and close colleagues. Such liberty also carries with it a responsibility to use Mailbox in a mature way. Wholesale broadcast messages, especially those unrelated to company business, are an imposition on the recipients, a misuse of resources, and bad manners.

Set up a routine for using Mailbox. The volume of your mail usually grows. Without a routine, messages can get lost or go unanswered. Having a routine also helps you avoid spending too much time in Mailbox to the detriment of your other activities.

Be aware of the diversity of terminals your recipients may be using. A growing number of narrow-width (80 characters) portable terminals are in use. Wide messages printed on such terminals are less readable than when printed on the sender's terminal. *EDIT* can be used to advantage to adjust the width of a message.

Mailbox is a tool, and customs and conventions regarding its use are evolving to meet human needs. Get to know its possibilities, as well as its limitations, and don't hesitate to experiment.

In summary, remember that Mailbox is only a tool. Used when appropriate with an awareness of its limitations, it is indeed a convivial one.

APPENDIX B

Summary of Programs, Actions, and Prompts

Program	Page	Purpose
ACCEPT	6	Removes received messages from active list.
ACCEPTALLBUT n	9	Removes all messages except those with numbers in n from active list.
ACCEPTED	21	Lists message numbers of accepted messages.
ACCEPTONLY n	9	Removes only those messages with numbers in n from active list.
DIRECTORY	19	Lists all enrollees whom you may address.
→GOON	4	Resumes an interrupted program.
INDEX 'x'	20	Returns information on a certain type of message, depending on whether x is *P*, *R*, *A*, or *L*, for pending, received, active, or lapsed.
LAPSED	21	Returns message numbers of accepted messages still in the Mailbox system.
ME	21	Returns your address code.
MESSAGE n	8	Returns the header and text of a message sent to you or by you. n is the message number.
MESSAGE 'abc'	8	Returns the entire message to □*NOTE* from abc. If abc is □*FILE* or □*NOTE*, it returns your message to □*FILE* or □*NOTE*, if you have one.
PENDING	21	Returns the numbers of all pending messages.
PREVIEW	19	Prints total number of pending messages, who sent them, and whether any are marked *URGENT*.
PRINT	6	Prints pending messages.
RECEIVED	21	Returns the message numbers of received but unaccepted messages.
REPEAT n	8	Prints messages sent to you or by you. n holds message numbers.
SEND	10	Begins the message sending process.
SENT	20	Returns information about who has and who has not received messages sent by you.
n TOFILE file	9	Copies messages with numbers in n to your file and returns pertinent information. file is a tied file's number or name, an untied file's name, or the name of a file to be created.
TRANSMIT x	10	Sends one message. x may be a variable or a character expression.
UNREAD	20	Equivalent to *PREVIEW* followed by *SENT*.
VERSION	21	Prints the identity and revision level of your copy of Mailbox.
WHOIS 'abc'	19	Identifies one person or group members by an address or group code abc. *WHOIS '≠abc'* and *WHOIS n* provide more information about user number n.
WITHDRAW n	10	Retracts one or more previously sent messages from recipients who have not yet received them. n holds the message numbers.

Actions within the *SEND*, *TRANSMIT*, and *WITHDRAW* Programs

APPEND n	15	Causes messages with numbers n to be appended for forwarding. */APPEND* discards them.
BCC codes	14	Directs copies to the addressees listed in codes, but does not include codes in message header. */BCC* codes removes codes.
CC codes	13	Same as *BCC*, but codes appear in header. */CC* codes removes codes.
CHANGE	14	Modifies message text.
CONF	12	Sets privacy level to *CONFIDENTIAL*.
DIST	14	Displays all individual address codes in your distribution list.
EDIT n	15	Composes message text into lines n characters wide.
MORE	15	Adds text to the end of your message.
NONCONF	13	Sets privacy level to *NONCONFIDENTIAL*.
PERS	12	Sets privacy level to *PERSONAL*.
PRINT	14	Displays message header and text that you have just prepared.
QUIT	15	Discards prepared message and distribution list.
REG	13	Designates a message as *REGISTERED*. You receive confirmation of its receipt. */REG* removes that designation.
SEND	15	Sends the message you have just prepared.
STATUS	14	Displays the message header only.
TO codes	13	Designates direct recipients of a message and adds them to the *TO* line in the message header. */TO* codes removes codes.
URG	13	Marks a message as *URGENT* to be printed ahead of all other messages. */URGENT* removes that priority.
WITHDRAW n	16	Retracts messages with message numbers in n from recipients who have not yet received them.
□	16	Activates the □: prompt for system commands or inquiry programs.

Prompts in the Mailbox System

TO	11	Enter distribution list, privacy level, urgency, and *APPEND*s, if any.
TEXT --	11	Begin entering or adding message text.
ACTION:	11	Add or correct entries from *TO* and *TEXT* --; then display, send, or cancel. If you make a mistake in responding to an *ACTION*: prompt, you'll receive another *ACTION*: prompt after the error report. Retype ignored actions.
□:	12	Reply with an appropriate expression.
...;	12	You have erred in answering the *TO* prompt. Retype ignored actions.
SPACE-CR	4	Exits the sending program, ends message text entry, or lists all possible actions.

INDEX

Note: An underlined page number indicates that comprehensive information about the item appears on that page. This index does not contain references to the section "Interpreting Reports from the Mailbox System."

Your views about this publication may help improve its usefulness; this form will be sent to the author for appropriate action.

READER'S COMMENT FORM

Possible topics for comment are:

Clarity Accuracy Completeness Organization Index Figure Examples Usefulness

MAILBOX

Cut Along This Line

Name and address (optional)_____

_____

_____

I do ☐ do not ☐ desire a reply.

Thank you for your cooperation. No postage stamp necessary if mailed in the U.S.A.

APL*PLUS MESSAGE PROCESSING SYSTEM
SAMPLE TERMINAL SESSION WITH COMMENTS

```
)[redacted]
OPR: APL CLASS IN BETHESDA; )LOAD 1 NEWS
532) 11.53.23 01/30/75 JQPUBLIC

  APL*PLUS SERVICE

      )LOAD 666 BOX
SAVED ...
      PREVIEW
2 MESSAGES PENDING FROM JZD WMS

      PRINT
30 JAN 1975  11.54.00
CONF, NONCONF: N
[1]
NO. 1553  SENT 30 JAN 1975  11.42.33
FROM WMS
TO   JQP

REMEMBER:  BREVITY IS THE SOUL OF WIT.

RECEPTION COMPLETE

      WHOIS 'WMS'
WMS    SHAKESPEARE, WILLIAM        BRIT DRAMA

      PRINT
30 JAN 1975  11.55.09
CONF, NONCONF: C
[1] CONFIDENTIAL
NO. 1535  SENT 29 JAN 1975  16.46.12
FROM JZD
TO   JQP
CC   NEWP WMS

NEW PEOPLE IN THE COMPANY HAVE NOT USED THE
MAILBOX BEFORE.  PLEASE SEND A MESSAGE TO
DEMONSTRATE THE SENDING PROCESS.  NEW PEO-
PLE ARE IN THE GROUP 'NEWP'.  THANKS. /JANE

RECEPTION COMPLETE

      SEND
TO NEWP
TEXT ---
HELLO, I AM SENDING YOU THIS MESSAGE TO
DEMONSTRATE THE MESSAGE-SENDING PROCESS IN
THE APL*PLUS MAILBOX SYSTEM.

TO PUT A BLANK LINE BETWEEN PARAGRAPHS AS
I JUST DID, PRESS RETURN AT THE END OF THE
LINE AS USUAL.  THEN, PRESS RETURN AGAIN.

AFTER ENTERING THE TEXT, I WILL MARK THE
MESSAGE CONFIDENTIAL, DIRECT CARBON COPIES
TO WMS AND JZD, AND USE PRINT TO PROOF THE
MESSAGE.  TO END TEXT ENTRY MODE, I'LL USE
SPACE-CR AT THE START OF A NEW LINE.

ACTION: CONF;CC WMS JZD;PRINT
CONF.
TO   NEWP
CC   JZD WMS

HELLO, I AM SENDING YOU THIS MESSAGE TO
DEMONSTRATE THE MESSAGE-SENDING PROCESS IN
THE APL*PLUS MAILBOX SYSTEM.
          .              .
          .              .
          .              .
MESSAGE.  TO END TEXT ENTRY MODE, I'LL USE
SPACE--CR AT THE START OF A NEW LINE.

ACTION: MORE
I AM ADDING A FEW WORDS TO THE TEXT WHICH
I FORGOT TO INCLUDE THE FIRST TIME AROUND.

ACTION: SEND
CONF. NO. 1590  SENT 30 JAN 1975  12.05.00
TO
ACCEPT RECEIVED MESSAGES
1 MESSAGE PENDING

      RECEIVED
1535  1553
      ACCEPT
2 MESSAGES ACCEPTED
```

Sign on to the *APL*PLUS* Service.
Special announcement from STSC.
Your port, time, date, your sign-on name.

To begin, load your Company Mailbox workspace.
Request a preview of your pending mail.

PRINT lets you receive mail at the terminal.
Current date and time.
Privacy prompt when there's *CONF* mail. Type *N* to print only *NONCONF* mail.
Message number; when message was sent.
Sender's address code.
Your address code.

The text of the message from *WMS*.

All *NONCONF* messages have been printed.

Find out who's who.

Use *PRINT* again to read *CONF* mail.

Choose *C* this time for *CONF* mail.
Privacy level as designated by sender.

WMS and individuals in group *NEWP* were carbon-copied on this message.
The text of the message from *JZD*.

All *CONF* and *NONCONF* mail has been printed.

SEND lets you prepare and send a message.
Designate main recipients: the group *NEWP*.
Prompt tells you to begin message text.
Type a line, then press RETURN to enter it.
Continue entering text, line by line, until you've said all you have to say.
Press RETURN twice to leave a blank line between paragraphs.

At end of this line press RETURN, type a space, then RETURN, to end text entry.
Specify new actions or change old ones.
The message is now printed at your terminal before it is sent.
Address codes appear alphabetically.

This is the start of the message text. To conserve space, not all of the text is shown here.

Action *MORE* lets you add message text.
What you type now is added to the text already there. Add as many lines as you like. Use SPACE-CR to end text entry.
Action *SEND* sends the message.
Message is filed, ready to be received.
Ready for next message. SPACE-CR to exit.
You've received mail but not accepted it.
A message has arrived; use *PRINT* to read it.

RECEIVED tells which messages you've read.

Incoming mail is saved after you've read it until you dispose of it by using *ACCEPT*.

Scientific Time Sharing Corporation
7316 Wisconsin Avenue, Bethesda, Maryland 20014 • (301) 657-8220

that are used to control the placing of cargo on ships. Via the hand-held terminals; dispatchers can route cargo in real time.

Lean inventory. In manufacturing, Mirecki says that supply lines have a natural need for portable radio/terminal systems as do control monitoring applications. For supply lines, the portable terminals replace the need to keep running to fixed computer terminals, and still allow a firm to control closely all raw materials necessary to feed the line. The payoff is in maintaining as lean an inventory as possible while keeping the line operating.

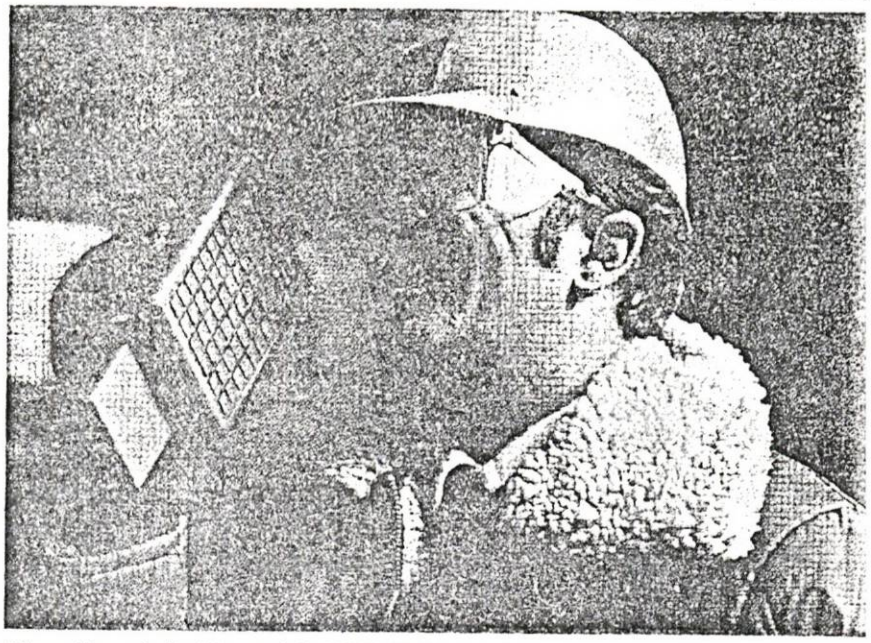
In the securities field, there is the potential to use these terminals on the floor of stock exchanges. Trades are now processed by hand in most cases, with overnight updates of each brokerage's computer, says Mirecki. Hand-held terminals could enable real-time ordering on the floor of a stock exchange.

Although this application has obvious promise, one Wall Street analyst suggests that there would be a great amount of resistance to such terminals because they would require the traders to use a keyboard, which is an unfamiliar instrument to them.

Meter maids. One field with the most natural need—law enforcement—is least likely to be first in implementing radio computer terminals. Meter maids who now must wait a day or more to spot an abandoned stolen car could use these terminals for direct access to a database of stolen vehicles. Parking tickets could also be entered on line. Such terminals could also be used for closer support of patrolmen pounding local beats.

Mirecki says that the Law Enforcement Assistance Administration (LEAA) has no development funds allocated to explore such a system at the moment. He has talked with police departments in Los Angeles and Kansas City only to find that although there is interest in the concept, neither department has the funds to investigate it further.

The payoff. Although the West German Post Office reportedly has estimated a need within Germany of 350,000 terminals (60,000 for industrial/commercial use) during the next six years—and U.S. vendors are talking about markets in the vicinity of $100 million per year—the approximate costs vs. benefits of the terminals are just now being determined.



More than talk. *Motorola's RDX-1000 hand-held, two-way rf radio terminal interfaced to mainframe computers for real-time data communications uses in inventory control, law enforcement, manufacturing, and other fields.*

A typical system has a price of approximately $70,000 for 10 terminals, which is a fairly expensive price when compared with hand-held batch computer terminals that are an order of magnitude less costly. Thus it would seem that the payoff for mobile radio computer terminals must be found in the increased productivity of employees and the value of real-time access to data, which takes a longer time to evaluate against direct cost-replacement systems.

"Evaluating has been an eye-opening experience for both us and our users," says Don Roch of Automated Systems for Distribution Management, a TauMark OEM that also markets fixed and hand-held batch terminals for inventory control. "Putting a dollar figure on the benefit, however, is difficult to do because of the detailed applications," he says. "How do you place a value on the system when it saves a supply line from 20 minutes' down time? But there doesn't seem to be any doubt that the value is there as long as the real-time connection to the computer is needed. The online connection is the key. We're talking with a number of Triple A Fortune companies, and they're all interested and willing to pay the additional money for the improved control."

Applications-oriented mail system opens on Telenet

The Telenet network has become the recipient of a sophisticated electronic message system that uses store-and-forward or mailbox techniques to deliver messages. Called Global Electronic Mail (GEM), the system was developed by Joshua Graham, a young financial executive who thought up the computer-based message-delivery system while working for the investment brokerage firm Goldman Sachs & Co. two years ago.

"I received a bill for message delivery and I thought it was outrageous," he explains, "so the Goldman people told me to develop a better system if I could do it." The

upshot is that Graham left Goldman Sachs to join New York City-based Corporate Time-Sharing Services (CTSS), which was formed by ex-Dartmouth Time-Sharing Services executives, who obtained a license for the operating system developed by Dartmouth.

Using the technical aid available at CTSS, led by senior analyst Mark Gillen, Graham developed the system, which now operates as a separate subsidiary of the company. GEM functions as a link between standard ASCII terminals and a host of public message-delivery systems such as Telex/TWX, Mailgram, or international Telex. The system also has the capability to force delivery to an auto-answer terminal or to function as a mailbox.

Custom carriage. The significance of the system, says Graham, is as much in the applications design as it is in the technical ability to deliver messages. Rather than just offer the system's inherent capacity to deliver messages, Graham and his staff of two dozen programmer/analysts are developing customized electronic message applications designed to fit the needs of the user of the system.

An example is a package that would link an automobile company to a list of purchasers. The system can handle broadcasting of messages to groups, he says, so that his service could be used for direct-mail marketing campaigns, recalls, or other needs. Another package, for the financial community, monitors cash management and automatically delivers cash availability and utilization data to the executives who need it.

The same set of data, says Graham, can be outputted in different formats based upon the need of the recipient. "The timeliness and usability of the information are obviously the most important factors in getting executives to use any data system," he says. "We customize a complete package for the user so that he is prompted through the transaction from message development to documentation."

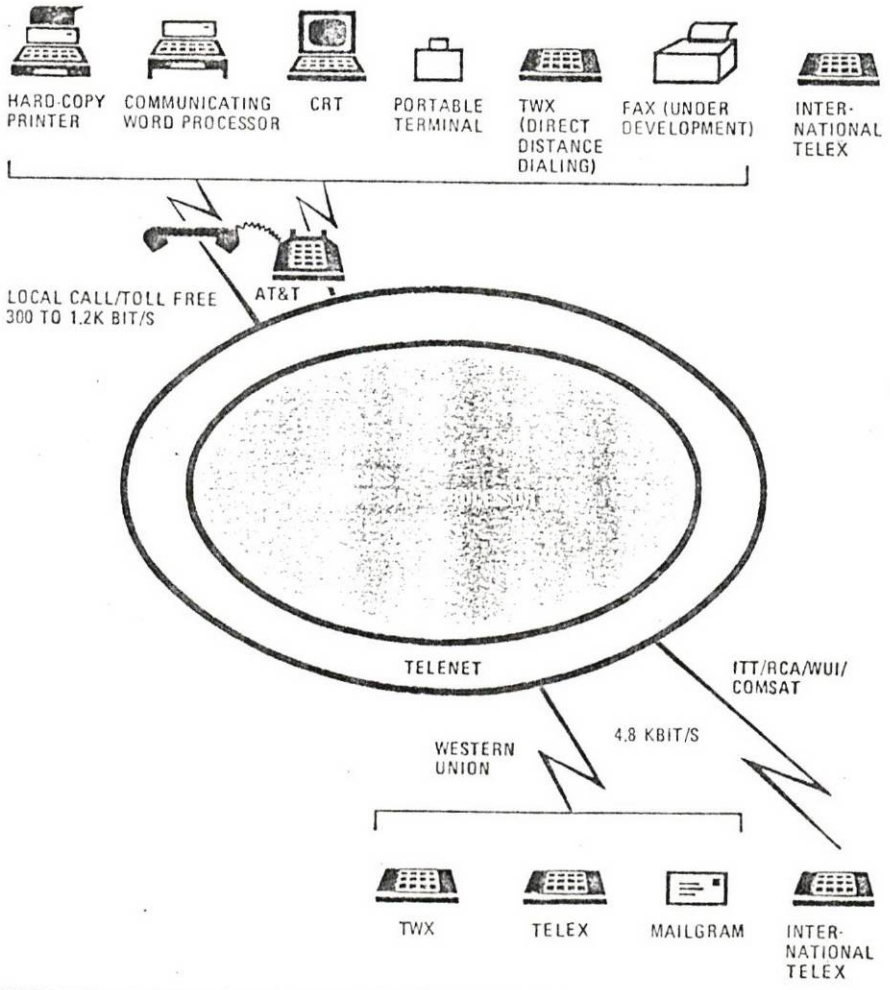
Because of these usability factors, Graham has customized his own terminals based upon Teletype Model 43s. The terminals, which Teletype maintains nationwide, have a special set of ROMs inside that automatically dial Telenet and establish the connection to GEM's computer. The user, says Graham, only sees the first prompt for his application, which can be a password for security.

Value-added. Graham's strategy—and it seems destined to be the cornerstone of the coming electronic mail generation—is to fit the technology to specific applications, making his service truly value-added. In this way, the merger between data processing and data communications is blurred beyond distinction. The Global Electronic Mail system, for example, isn't even a regulated carrier, but a computer timesharing service, says Graham's lawyers. He also points out that all physical communications is handled either via the Telenet or the switched analog networks.

It is this inextricable merger of computing and communications that Graham, as well as a growing list of industry observers, sees as paving the way to hundreds of unique applications. Ken Bosomworth, president of International Resource Development, points out that the end result of such development is the "personal business terminal," which, he says, will sit on the desk of a white-collar worker to perform customized processing and communications tasks. An airlines reservation terminal, he says, is just such a terminal, as is the cash-management terminal

Electronic mail

Integration. *The Global Electronic Mail system integrates timesharing-style computing with message handling to produce a value-added, computer-based electronic message system via Telenet's public packet network.*



developed by Graham.

Although the user community is not yet flooded with these personal terminals, Graham says it is overflowing with potential applications for GEM, whose prices are based upon the number of messages transmitted. Graham's rate is about $1.50 per message, which is expensive when compared with a simple computer mailbox system, but cost effective when his system is used to provide otherwise unavailable data, Graham explains.

In addition, he says, there are international applications with huge savings. The current cost for an international telegram to the U.S. is about $16, he says, which can be enormously expensive when messages are broadcast to multiple addresses at that price. By sending one telegram to GEM, he says, users can reduce the cost of additional telegrams to his price of $1.50 for a Mailgram.

Xerox backs off subminute fax

One of the long-standing points of controversy in the facsimile field is whether Xerox, the industry's leader in terms of installed base, is going to enter the fast-growing subminute digital facsimile field. Nobody doubts that Xerox has the equipment. In fact, the company reportedly has two machines—one developed in the U.S. and another by Fuji Xerox in Japan.

Enter naught. One leading expert in facsimile, president Howard Anderson of the Yankee Group, has predicted Xerox's entry on numerous occasions, based upon leaks from his sources. Somehow, however, each occasion has always come to naught. Most recently, the company was expected to announce its subminute machine at a major trade show. "They never made it," says Anderson. "I understand that there were some internal problems about the pricing and the machine was never announced."

The interest in Xerox is more than academic. From a business with one vendor—Rapicom, in 1977—the digital facsimile field has grown dramatically, with Graphic Sciences, Minnesota Mining & Manufacturing (3M), Panafax, and Southern Pacific Communications all entering it. In addition, Nippon Electric, Mitsubishi, Sharp, and at least two other Japanese firms are known to be eyeing entry into the $50 million field that is "growing at 40 percent annually," according to Anderson. There is great concern that Xerox would clog an already crowded field.

Future links. Digital facsimile is often considered to be more flexible than the less costly, but slower, analog facsimile machines that currently dominate. The reason, says Rapicom's manager of networking, David Mack, is that digital facsimile can be integrated into corporate networks in simpler fashion. Rapicom, he says, already has a system whereby the firm's digital facsimile machines can be linked to IBM mainframes as part of an overall corporate network.

Coming down the pipe, however, is a system from Compression Laboratories of Campbell, Calif., that could change the lead enjoyed by digital facsimile machines. The small data compression firm has developed a complete store-and-forward facsimile system that can receive data from analog facsimile machines, convert it into a compressed digital format, and retransmit it to another machine.

In addition, International Telephone & Telegraph says it is near the end of its long and difficult road designing its Faxpak network that will link incompatible facsimile machines. ITT recently filed its tariff for the Faxpak service, which will be priced about 40 percent lower than AT&T's public switched telephone network so long as the user is willing to accept a delay of up to four hours between the time the message is sent to ITT's store-and-forward network and delivered to the designated recipient.

Uncertain impact. The advent of the Faxpak network, along with hardware from Compression Labs and others, could have a moderately negative impact on the use of more expensive, but easier to use, digital facsimile systems. The reason, says one observer, is that users who might have switched to digital facsimile from analog machines now have less costly alternatives. In this observer's opinion, however, there is enough growth in digital facsimile to make that potential user segment only moderately important.

"Digital facsimile is used in corporate telecommunications center for the most part," he says, "while analog facsimile tends to be used as convenience devices inside offices. Faxpak will pull its traffic primarily from the convenience segment, which now uses the public switched network. Although some digital facsimile users will switch to Faxpak, a larger number will remain with leased lines."

As far as Xerox's entry into the field is concerned, the observer believes that the uncertainties in the digital facsimile field are probably enough to keep the firm out of the market. "While the market is undergoing rapid growth," he says, "it is not certain that this growth will continue. In the last recession, for example, Xerox was severely stung in its facsimile business. The company has not introduced a new product since that time and may have some real doubts about introducing one in the face of another recession."

Rugged keyboard weathers abuse

A small West Coast company has introduced a ruggedized, self-enclosed, solid-state, membrane-type keyboard that is likely to open up new vistas for communications terminals. The keyboard is completely free of moving parts and can operate in such rough or dirty environments as grain elevators, machine shops, military tanks, and offshore oil rigs.

In addition, its low price could make it an inexpensive control panel for executive suites or the home hobbyist if cybernetic factors can be overcome. Developed by Touch-Activated Switch Arrays Inc. (TASA), the keyboard will be ready for quantity shipments in a few weeks, according to TASA market-

ARPA ~~SRI~~ MESSAGE SERVICES MEETING #1 1/10/79

Proliferation of services

new things coming

multimedia

FAX

internetting

DCA picking up technology

set stage for ARPA taking a more active role in setting standards, etc.

DALE FARBER: status of different msg systems

(he sent out a questionnaire, got some responses)

(many people don't know where their message syster came from)

general impression - most message systems were not sponsored or funded to be maintained.

Vint says most AN software was developed informally - no official responsibilites in the past - no tasking in proposals

(#5) <CAHCON> SURVEY.TXT

PROBLEMS

TED MYER 2 classes of problems:

#1 - shortomings in delivery servic
mail problem - some solved by mail/file operation.

#2 format (msgs) changes faster than message system can keep up with.
Hermes can't parse all of #3?.

MSG - in limbo - at RFC 680 level (preceded 733)
both generate compatible msgs since 733 backward compatible with 680.

TENEX system doesn't put host names on addresses for local mail. This violates 733.

733 was a minimal step forward toward standardization, with assumption that ARPA could enforce standardization.

Farber: A problem - users want to modify their own receivers - rather than complain to transmitters - when incompatibility arises.

Two kinds of sites -

HOST TABLE PROBLEMS (UPDATING)

NSW problem - "hostless users" virtual hosts, not bound to any particular host.

no network-wide msg-transmit systems.

Authentication - TENEX. - spoofing - efficiency.

MESSAGE ID problems.
no duplicate filtering.

~~Two~~ two conflicting goals: < headers people can read
headers machines can read

Adams - not inclined to redo 733
look for future developments.

Efficiency - A msgs → N copies; N connections

Limit inflow of messages.

MAIL FILE - advantages

unsolicited stuff because FTP uses a terminal socket to log in. [must strip down etc]

mail systems Very FRAGILE - no protocol(!)

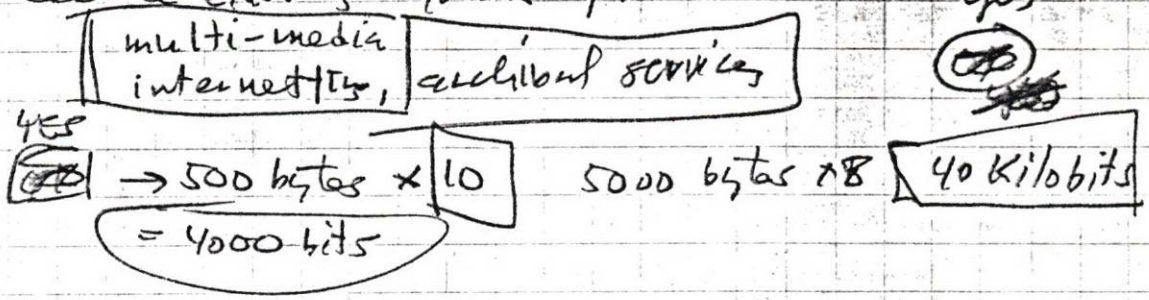
Much AN software (FTP, TELNET) assumes a human being at the ends.

need to re-think the problem, small group of people, come up with a design.

Need more than 1 protocol at once.

Not Clear Just one protocol can be simple / complex.

Farber: → declare existing system PROTEA.



alternatives: try to get everybody to do MLPL

---

3 MAJOR ISSUES multimedia, internet, archival

ARPA - (Vint) → not comfortable spending much $ on lashing different msg systems together

(Adams to be in charge)

UCL Dacom FAX, 8050 send FAX data to data computer.

AI
LCS: LISP MACHINE plans to use a shared-file system.
mail stored on shared-file system.

WORD PROCESSING INDUSTRY has formed a standards group.
(Debbie is a member)

Bob Thomas BBN — NSW Mail Servers

Constraints — no mail system R&D
— make use of existing AN software
\ interoperability NSW ⇔ AN

Approach
- provide NSW software segment for AN-like mail
- install existing mail systems as NSW tools
- Use existing msg. delivery systems

POSTEL — The way AN handles mail is all wrong.

(1) NO FTP
new mail send/rcv servers. ("Post Offices")

(2) New mail format: structured, machine-oriented
Like NSW-BB or MSDTPt (Jack Haverty's)
↑ F bit
Lists of items, semi-implied ((addr, body), (addr, body), ...)
TYPED data structures

(3) New user programs - to send messages in these formats
implies that elements are displayed in a different form
(transformation)

(4) HACK slightly modified SNDMSG ⇒ allow internet addresses.
extra line in header ⇒ real internet address.

Adams: Auto dialing — volume of traffic insufficient to justify dedicated AN connection.
simply
use a dial FTP to transfer mail from

Farber: DARCOM, U. Delaware,
remote site to an AN system.

(1) user in disconnected host
(2) name of disconnected host
(3) routing

charging/acc

Hawley: (ANSI) standards for message systems just starting:
(1) MSG STRUCTURE header
(2) MSG DELIVERY (protocols)
(3)

proposed IFIPS WG 6.5 "Int'l Msg Services" common carriers

just setting started. French, Engl, Canadian

OTA Working Group/Panel on Message Systems

3 to 5 committees chartering this OTA group.

Duane Adams' Summary. (3:15 pm)

Goal: stabilize (don't tinker)
(A)
1. No new things added to 777
2. Msg systems not to add new features, except those in 777
3. ask few vendors to remove incomp. features
4. ask vendors to fix problems

(B) Near term Internetting & fax, not yet clear; ARPA to resolve. & (INWG)

(C) Outstanding problems: (1) local net problems
(2) networks coming up in Europe this summer; also Ft Bragg PRNETS.
(3) may be an interim post office forwarder

(Where do we go from here)

Duane: 3 Areas needing work

N.B: Postel's PO is functional replacement for MAILER

(1) Addressing Issue (user level, machine level)
- Internetting address 8 bits NETID, 24 bits HOSTID / POST & PR.

* Invite 2-3 page position papers on addressing by 1 March (incl. user interface)

(2) Format & Structure of a Message: will mail out Postel's notes ≈ 1 March for comment

(3) System Architecture & Distribution –

* Soliciting views by 1 April on architecture

end, 3:30 pm.

Guests for Message Service Meeting

January 19, 1979

Bolt Beranek and Newman,
Cambridge, Massachusetts

ARPA: Lt. Col. Duane Adams (Adams at ISI)
APPA: William E. Carlson (Carlson at ISI)
ARPA: Vinton G. Cerf (Cerf at ISI)
BBN: Jerry D. Burchfiel (Burchfiel at BBNA)
BBN: Debra P. Deutsch (DDeutsch at BBNA) → FAX
BBN: Douglas W. Dodds Jr. (Dodds at BBNA)
BBN: John F. (Jack) Haverty (JHaverty at BBNE)
BBN: Charles A. Khuen III (Khuen at BBN-TENEX)
BBN: Mark A. Lavin (MLavin at BBN-TENEXE)
BBN: Charlotte D. Mooers (Mooers at BBN-TENEXE)
BBN: Theodore H. Myer (Myer at BBNA)
BBN: Raymond S. Nickerson (Nickerson at BBN-TENEX)
BBN: Paul J. Santos Jr. (Santos at BBNE)
BBN: Robert H. Thomas (BThomas at BBND)
BBN: Michael A. Wingfield (Wingfield at BBND)
CCA: Joanne Z. Sattley (JZS at CCA-TENEX)
CMU: Howard D. Wactlar (Wactlar at CMU-10A)
D.O.E.: James R. Pool (Pool at BBN)
DCA: Robert MacNab (DCACODE535 at ISI)
DCEC: Edward Cain ()
DCEC: Warren Hawrylko () ←
DCEC: Harry A. Helm (Lyons at ISI)
ISI: Dan Cohen (Cohen at ISIB)
ISI: Jonathan B. Postel (Postel at ISIB)
MIT-AI: Patrick H. Winston (PHW at MIT-AI)
MIT-DMS: P. David Lebling (PDL at MIT-DMS)
MIT-DMS: Albert Vezza (AV at MIT-DMS)
OFDA: Wayne Shiveley ()
Rand: Robert H. Anderson (Anderson at RAND-UNIX) → MN
SRI: Kenneth L. Harrenstien (KLH at SRI-KL)
SRI: Ronald C. Kunzelman (Kunzelman at SRI-KL)
U. Delaware: David J. Farber (Farber at OFFICE-1)

John — Dennis DW.H. Paul TFS Phil (last) —

Newsfront

Coming soon. After much internal tugging, AT&T's new BDN net is beginning to fall in place, sources say.

AT&T is preparing national data net with easy access

Bell to build around CCIS and ESS switch with standard protocols; FCC approval needed

One of the worst-kept secrets in the data communications world is that AT&T is planning a nationwide digital data network called Bell Data Network (BDN). While no one—including top AT&T officials—knows precisely what BDN will look like when finally implemented, enough information is leaking out to be able to detail the major characteristics of the network.

The communications giant has abandoned its earlier plans to design the network around large mainframes supplied by outside vendors and has elected instead to follow the time-honored Bell System's method and build the major components of the network itself. The new net will be built around Bell's No. 4 ESS switch and will utilize Bell's common channel interoffice signaling system (CCIS). The latter is a packet-switched network that transmits signal control information between ESS stations.

At customer sites, Bell is planning to offer a means of interconnecting the network via standard protocols. Plans to design a proprietary AT&T protocol for the network have been abandoned. The customer site plans, however, remain a moving target because they depend on the latitude that the Federal Communications Commission will give AT&T.

The Bell System is said to be preparing to file a tariff with the FCC within six months, but a true nationwide network could not be a reality for at least 18 months, because it will take at least that long to install the No. 4 ESS switches in at least 50 cities—a number that would have to be reached before the network would be considered to be approacning nationwide status. Twenty No. 4 ESS switches will be installed by the end of 1978.

Victory. The BDN project has met with much contention within AT&T among various Bell System units, and, more than anything, it represents a victory for AT&T's marketing operations unit in Morristown, N. J. Headed by former IBM executive Archie J. McGill, now AT&T's director of market management and development, the Morris-

town division directed the design of BDN and it represents the triumph of the marketing unit over AT&T's research arm, Bell Labs.

It was just over a year ago at a meeting of the Communications Division of the Electronic Industries Association at Hyannis, Mass., that McGill said AT&T was "moving from a technology-driven corporation to a market-driven corporation."

AT&T headquarters in New York City had no comment on BDN.

Vendors contacted. Over the past several months, AT&T has contacted several computer and hardware manufacturers with the idea that their gear might be designed into a nationwide digital network that would offer a proprietary AT&T-designed protocol. At one point, large-scale Honeywell mainframes were said to have been under consideration as network switches.

The final choice—the No. 4 ESS switch—is a large capacity solid-state switch that utilizes a common control system and time-division multiplexing. Although originally designed for voice communications, the switch has capacity large enough to handle BDN. Computer Communications Inc. of Torrance, Calif., has prepared a front-end processor design for BDN and that company is considered to be the front runner to make front ends for the network.

Transmission specifications call for a wide range of popular speeds, although the offerings would be no match for those of Satellite Business Systems. The Bell network will be initially aimed at low volume and small users, while SBS is directing its services at the high volume data communications traffic of the largest corporations in the country.

BDN does not represent so much of a technological jump as it does a change in marketing concepts and, indeed, BDN's preliminary specifications do not look like much of an improvement over AT&T's existing all-digital network, Dataphone digital service. Like DDS, BDN will not be an entirely new network, but rather will consist primarily of improvements in AT&T network nodes.

BDN's customer site equipment has been the subject of spirited debate within AT&T and is likely to continue to be, even after the network is publicly announced. However, a source within AT&T notes that a major issue has been settled in that the network will be easily interfaceable with most non-AT&T terminals, including IBM terminals. The source added, though, that AT&T will probably stop short of an outright adoption of the X.25 standard protocol as BDN's protocol. Earlier, there had been some thinking within the Bell System that an AT&T proprietary protocol that interfaced solely with AT&T terminal equipment would be the best way to go with the network.

Customer controllers. Nevertheless, AT&T has not given up on the idea of supplying its own customer site controllers. Several months ago, AT&T marketing director Roger Moody said the Bell System would supply a customer site controller for future "communications processing" applications, and he mentioned the Dimension PBX specifically. At the time, Moody noted that the Dimension was capable of interfacing with a variety of office data terminals. In addition, reports continue to filter out of AT&T that the firm is preparing another customer site controller designed around Digital Equipment Corp.'s LSI-11 microcomputer.

There is good reason for indecision on the customer site controller issue—the FCC. The Commission is currently in the midst of its Computer Inquiry II, which will define the line of demarcation between data communications and data processing.

An FCC decision negative to AT&T on the issue could mean that Bell would be forbidden to offer customer site controllers.

Industry loses a pioneer

Every business has a father or two and data communications is no exception. One of the field's great pioneers died last month at his Connecticut home not too long after his 101st birthday, and it would be an understatement to say that the pioneer, Edward E. Kleinschmidt, had a rich and full life.

Founder. *Edward Kleinschmidt, inventor of the Teletype, with his multiplex printer and type printer.*

Kleinschmidt's imagination was such that he might best be described as an obsessive inventor. His masterpiece was the Teletype transmitter-receiver, and millions of them were made. On the side, Kleinschmidt invented toys, the automatic fishing reel used by sport fishermen, and the high-speed ticker tape transmitter used for following the stock market.

Born in Bremen, Germany, Kleinschmidt came to the U.S. when he was nine, and by the time he was 16 he was peddling his first invention, a battery-powered pinwheel. Unfortunately, that flopped when its acid batteries proved leaky. Self-educated, he went back to the library to learn more about electricity.

An early interest in telegraphy led him, in 1902, to invent a facsimile machine which failed because it was too far ahead of its time. In spite of its failure, that invention eventually led, 12 years later, to the introduction of the Teletype machine. By merging with his only competitor in 1928, he founded the Teletype Corp. AT&T bought him out in 1930, but of the $31.5 million AT&T paid for the company, Kleinschmidt got just $150,000.

President's Plane. He subsequently left Teletype and formed Kleinschmidt Laboratories in 1931,

continuing his work in electronic communications. In 1948, the U.S. Army Signal Corps accepted his 100-word-per-minute printer as a standard device for use in combat conditions, and the unit worked well enough to be included as part of the communications equipment aboard the President's Air Force One.

Kleinschmidt Laboratories was acquired by SCM Corp. in 1956, and Kleinschmidt, then 80, retired again—and again he continued to invent. In 1966, SCM introduced Kleinschmidt's Telescripter. It was a high-speed teleprinter about the size of a car radio, designed primarily for use in police cars and commercial truck fleets.

A soft-spoken man, he was nonetheless considered a difficult taskmaster by those who failed to move projects fast enough for him. He also had a healthy dislike for resting on his laurels. His workshop is said to have contained nothing of his past achievements—only a complete set of tools for the project currently at hand.

Experimental fax net handles NASA teleconferencing

For the past two years, teleconferencing has been a major application of an experimental switched 55-terminal facsimile network operated by NASA and supported by terrestrial circuits. Through remote stations equipped with voice-actuated microphones and facsimile equipment, the network has enabled some attendees to participate without having to travel to the meeting place. NASA has summarized the results of this teleconferencing project in a report.

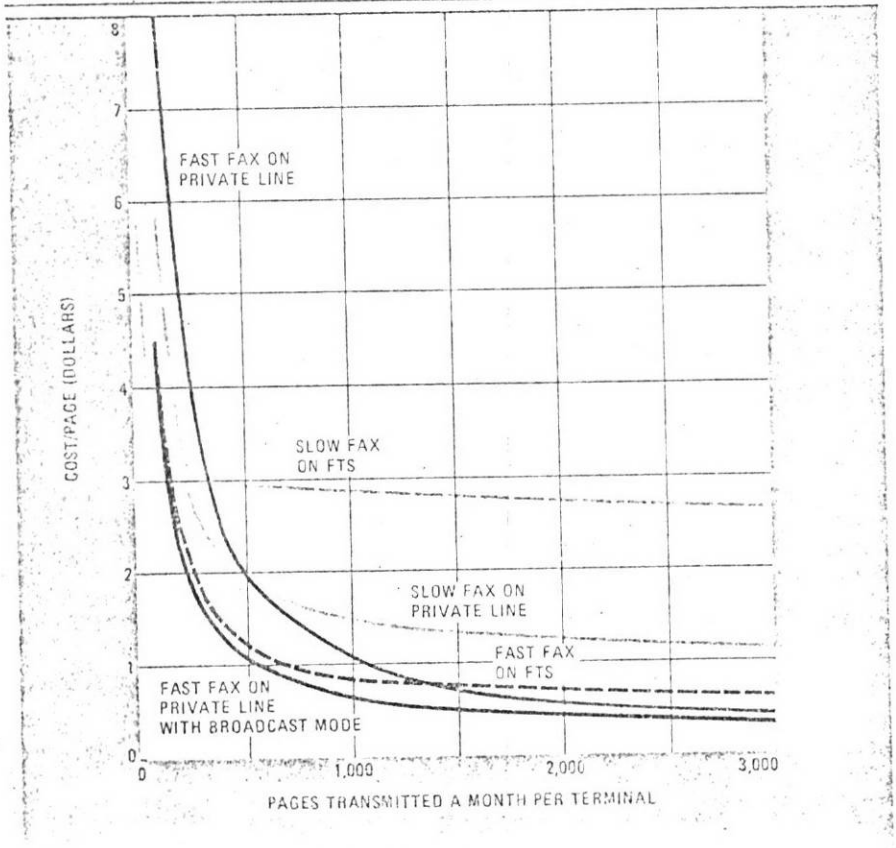
Two different facsimile systems were used in the teleconferencing demonstration: one employed Rapifax terminals that scan, transmit, and print a page of text in 40 seconds to one minute, while the other consisted of Magnafax equipment that processes a page in 4 to 6 minutes. The Rapifax machines operate in broadcast, as well as point-to-point mode, and this has a bearing on costs. So does the number of unattended fax stations.

The conclusions, based on a total of 687 meetings, included an estimate that teleconferencing had avoided travel costs amounting to $3.3 million, or about 21 percent of NASA's travel budget. The annual cost of the teleconferencing network was about $500,000.

In the accompanying figure page costs and speeds are plotted for the two types of terminals, using FTS (federal telecommunications system) and private-line circuits. NASA pays about 14 cents/minute for an FTS circuit, and an average of $475 a month for each private line utilized in the test.

The average fast fax terminal in the teleconferencing test sent 863 pages/month. The average cost/page was 63 cents. Including the paper, ink, and labor costs, this average cost/page figure increased to about $1.10. Assuming half of the received pages were broadcast, and half of the transmitted pages were sent to unattended terminals, the average cost decreased to 71 cents/page.

NASA facsimile test



Money-saving network. *The chart illustrates that, at 2,000 pages a month, private-line fast fax in broadcast mode is cheapest, followed by private-line fast fax, fast fax on FTS, slow fax on private lines and, at top, slow fax on FTS.*

Satnet brings costs to earth

An experimental program, being conducted by the Department of Defense Advanced Research Project Agency (ARPA), linking the U.S. and Europe in a satellite packet-switched network, may be a prototype of future international data communications nets.

The program, labeled Satnet (satellite network), links satellite earth stations in West Virginia, England, and Sweden in a 64 kbit/s data communications network, via the Intelsat IV-A satellite, using one channel in a multi-access arrangement. Because a multi-access, single-channel configuration conserves satellite transponder use, slashes transmission costs, and is ideal for teleconferencing, its use may make it easier for developing countries to implement an international data network.

"If a country purchased a single channel," says F.E. Heart, vice-president at Bolt Beranek and Newman, the Massachusetts-based consulting firm participating in the ARPA program, "it would save the enormous cost of buying many dedicated point-to-point channels that is necessary today to have an effective international data link."

One channel. In a conventional satellite communications network, each station is assigned a separate up-link frequency, or channel, for transmission and the satellite responds on another separate down-link channel. Satnet's multi-access network, on the other hand, employs only one frequency for all participants and uses a combination of packet-switching architecture and an earth station satellite information message protocol (SIMP) minicomputer, which stores a packet communication until the channel has sufficient bit space available to transmit it. When the packet space is available, the SIMP minicomputer turns the earth station transmitter on and the packet is sent to the satellite.

On the receive end, the SIMP minicomputer reads all the satellite-transmitted packet headers, which contain the address codes, determines which message is for its earth station, and blocks the others. Because all the packets arrive at all the earth stations at essentially the same time, full network teleconferencing is only a matter of manipulating the header address codes to include all the stations.

"Teleconferencing is one of the key applications of Satnet," says Robert E. Kahn, ARPA's deputy director of information processing and chief scientist, "especially since it is so easy to address all the nodes in the network at the same time."

The SIMP minicomputer is a Honeywell 316 with a 32K word memory, which is tied directly to the earth station. A unique aspect of the SIMP minicomputer is that it can be accessed with low-speed land lines. The SIMP minicomputer multiplexes the input data and passes it on at 50 kbit/s to the earth station, which then transmits it at 64 kbit/s. This, according to Kahn, makes the speed of the land-lines far less important than in a conventional network.

Maximum channel use. The satellite experiment, which has been on-line for about two years, has tested many protocols in an attempt to maximize the usage of the satellite's single channel, but the most recent candidate—contention priority oriented demand assignment (CPODA)—seems to be favored by the scientists.

"The CPODA protocol will get us very close to the theoretical maximum channel use," says Kahn, "and, it's particularly attractive because it lets individual users select a delayed service at a reduced cost."

This is possible because CPODA has a capability that allows the user to assign a degree of urgency and a time window for his transmission. If, for example, a message were originated in Europe, but could arrive in the U.S. anytime within the following 48 hours, it would be stored in the SIMP minicomputer to await a reserved transmission time, selected by the SIMP mini-

---

Southern Pacific Communications (SPC) and Tymnet have agreed tentatively to **interconnect their facilities with the new Canadian-National-Canadian-Pacific (CNCP) Info-Switch network.** Service is expected to begin before the end of the year. Initially, U.S. customers will be connected with the Canadian carrier's Info-Exchange and Info-Call services. The former is a digital, circuit-switched, private-line network; the latter provides limited (node-to-node) packet switching. Access to Infogram, CNCP's fully packetized network, is planned later. One of the first users of the new combine is expected to be the Ford Motor Co., which plans to transmit facsimile messages at 9.6 kbits/s among eight terminals in the U.S. and Canada.

Data may follow a simpler voice route if a large timesharing vendor implements the proposal of TDX Systems of Vienna, Va. TDX has proposed the installation of its **microprocessor switches, claiming they will reduce the number of phone lines needed by dial-up terminals,** since the switch will automatically route the connecting terminal to its proper modem at the host site. At present, separate lines are needed for terminals operating above 300 bit/s. This represents a new application for these switches, which previously have been used mostly for voice operations.

After getting off to a fast start in the early 1970s, **sales of independently supplied modems are beginning to slow** in their penetration of the growing telephone interconnection market in New York State. Independent manufacturers now have about 37 percent of the modem market in New York, according to a 1977 update of a 1975 study on interconnection by the New York State Public Service Commission. However, the report, which is based on figures supplied by New York Telephone Co., indicates that the situation hasn't been all bad either for Bell's datasets, whose growth has increased by 26 percent. Thus, while the independents are solidly in the market, users will continue to have a choice of modem vendors, ranging from a host of independent suppliers to Bell itself. "It can be seen," states the report, "that the growth of interconnection has stabilized in all service categories, with the rate of growth declining slightly."

computer on the basis of least satellite traffic. This arrangement, Kahn says, could lead to a flexible pricing structure, with users who are willing to take slower service paying less than those demanding faster communications.

While the technology aspects of the satellite network look "extremely encouraging," notes Kahn, adoption of the concept by the international satellite carriers is less than certain. Although the multi-access network would likely be applied in low-traffic situations and would be unlikely to impact large commercial users, the carriers are reluctant to introduce a pricing structure which would be based on use rather than on the number of leased circuits.

Swift prepares on-line U.S. service

More than 60 U.S. banks were preparing to go on-line with the Swift international banking network later this month as the Europe-based network is getting ready to inaugurate the service in the U.S.

About 175 European banks—and several European branches of U.S. banks, too—have been connected to Swift since the service was switched on in Europe in May. Volume is about 4,100 messages a day.

Many of the problems that delayed the activation of Swift for almost a year (DATA COMMUNICATIONS, May 1977, pp. 13-14) appear to have been ironed out. The Burroughs equipment in the two main European switching centers has been upgraded, with the result that one U.S. bank member of Swift said the Swift officials are convinced the network will have little difficulty handling its more than 500 member banks.

However, the Swift interface device (SID) represents a different story. The status of the terminal module is described as still not stable and the two U.S. manufacturers of SIDs—Burroughs and General Automation—are still working to improve their offerings.

Bandwagon. At the same time, other vendors without Swift SID contracts were increasing their marketing efforts for their gear, which, since it is not endorsed by the Swift network and cannot carry the Swift name, is called computer-based terminals (CBT).

IBM, Univac's Varian Data Machines, Arbat Systems Ltd., and NCR were all actively marketing interface equipment that is functionally the same as a SID.

Laurence E. Wadman Jr., a Chase Manhattan Bank vice president and a U.S. director of Swift, said the network has been testing in the U.S. and Canada an advanced control function that includes a capability for alternate routing in the event of service interruption. That function, he said, is scheduled to be adopted through-

out Swift, beginning in October.

Meanwhile, Swift has filed a protest with the European Economic Community over rate increases ordered by CEPT, the consortium of European government monopolies that controls the various European national postal, telegraph and telephone agencies called PTTs. The PTTs unexpectedly jumped the rates they charge Swift customers and the network is officially protesting the increases. In its complaint against the PTT consortium, Swift has charged unfair predatory tariff-fixing practices.

There are some fears that the PTTs' pricing actions could trigger others to seek increases. In a report on "New Trends in International Telecommunications" prepared by the Center for Communications Management Inc. of Ramsey, N.J., the issue is raised.

"One should not get the impression that it is only the European PTTs who are seeking to impose these punitive tariffs," the report states. "The U.S. international record carriers are hot on their heels.

"They are also proposing volume tarification for Swift, based on public Telex rates. But unlike CEPT, they are suggesting not 10-15 percent of the public Telex rate, but 30 to 35 percent. They justify these high rates by saying that they have estimated that the U.S. record carriers will lose $20 million."

Electronic mail: where is it going?

Does electronic and computer mail have a viable near-term future or is it just a fleeting gimmick? With the thought of obtaining an answer to that question, a telephone call was placed from New York City to Menlo Park, Calif., to Raymond R. Panko, who prepared an exhaustive report on "The Outlook for Computer Mail" for SRI International.

"Mr. Panko is in London," said his secretary at the technology research organization, "but I'll just put the message over the Arpanet system for him. He checks his electronic mailbox a couple of times a week no matter where he is."

Without intending to, Panko's secretary answered the question. Indeed, evidence is building that electronic and computer mail is catching on, although some troubling questions about the issue are surfacing, too.

"Many trends may be converging," writes Panko in his report, "and these could spur the evolution of computer mail. But, to keep things in perspective, it must be borne in mind that all computer mail's prospects are highly speculative. Business and government users have yet to really probe the adequacy of current design notions, and near-term developments are simply impossible to predict."

Running the gamut. Users are beginning to experiment with their own electronic mail systems and, as they do, it is becoming evident that various users are approaching the concept from vastly different directions—the technologies run the gamut from packet switching to communicating word processing.

Money talks, too. And when it does in the context of electronic mail, it says that costs are becoming competitive with other more traditional ways of sending messages. "Many corporations," writes Panko, "have found that they could communicate across the nation more economically by writing on and reading a common file accessed via a timesharing network than by using Telex or TWX."

Citibank in New York has several minicomputer-based word processing workstations interconnected and, although a bank spokesman calls the system "very experimental," he says the bank is "definitely committed" to moving ahead with the effort.

Robert B. White, an executive vice president responsible for implementing the systems at the bank, has described the versatility of the system. Like many other electronic mail configurations, the Citibank workstation has a keyboard CRT and hard copy printer as key workstation components.

An individual's electronic mailbox can be accessed by simply keying in at the CRT. Appointments, phone messages, internal memos, and postal mail can be easily displayed

Deceiving look. *IBM's System 6, with its 6640 Document Printer, performs electronic mail chores.*

on the CRT. Messages can be communicated at 1.2 kbit/s over a dial-up voice network and, when needed, printed out on-line at the workstation. Plans call for the Citibank system to have a Basic programming capability.

Citibank is also connected with the packet-switched Hermes message system, which is partially sponsored by the Department of Defense Advanced Research Projects Agency (ARPA). Designed by Bolt Beranek and Newman (BBN) Inc. of Cambridge, Mass., and offered through Telenet's public data communications network, Hermes is nearly universally available because of easy accessibility.

"For instance," says John M. McQuillan, manager of BBN's system analysis department, "much of BBN's contact with Citibank is on-line. I've been using Hermes at BBN for three years now and I take it for granted. I recently wrote a 65-page report with five other people and we never talked with one another. The writing and editing were all done on Hermes."

Mail filing. McQuillan points out that the Hermes message system is handy for filing, too, adding that he "cleans up" his messages once a month and electronically files them away in memory for possible future use. For terminals, McQuillan has a Texas Instruments' Silent 700 printer and a Teleray keyboard-based CRT, although he notes that the system can use most popular terminals. BBN designed the electronic mail system around a Digital Equipment Corp. PDP-10.

Many data processing vendors are working on electronic mail systems, but their efforts appear to be following a crazy quilt pattern with various vendors approaching the problem from vastly different equipment perspectives.

Telenet and Tymnet, for instance, are developing and refining electronic mail systems based on their existing packet-switching technologies, while large equipment manufacturers, such as IBM, Xerox, and Digital Equipment are thought to be developing electronic mail gear based on existing equipment.

Xerox's strength in the digital facsimile and telecopier market makes a logical encore in the electronic mail area a possibility.

While IBM has not heavily promoted its efforts in the electronic mail field, the firm's 6640 document printer, an offering within the Office Product Division's System 6 product line, definitely looks like electronic mail equipment. The 6640 can communicate with computers and with other 6640s over voice-grade phone lines. At the other end, output is printed at rates up to 92 characters per second.

Prototyping now. Howard M. Anderson, president of the Yankee Group of Cambridge, Mass., a data communications marketing research firm, believes vendors' in-house prototype systems of today will look like the systems of tomorrow that will be offered by vendors in the marketplace.

"It's primarily a cost thing," says Anderson, who recently made a survey of electronic mail trends. "The right economics are there now for information processing manufacturers to have their own experimental electronic mail systems. And, as long as hardware costs keep dropping, the economics will be right for the manufacturers' customers in a few years, maybe sooner."

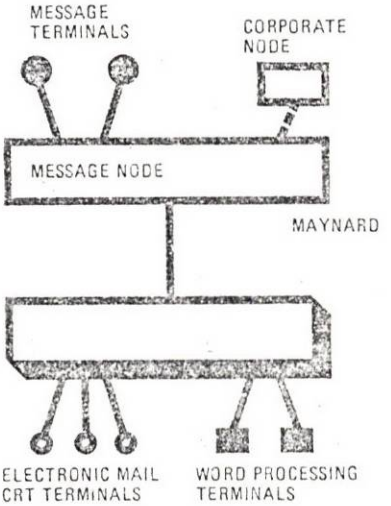
If Anderson is correct, then there may be a glimpse into IBM's future in its electronic document transfer system that connects five scattered IBM facilities in the U.S. One of the locations is IBM's Office Products Division in Franklin Lakes, N.J., and much key equipment involved in the system is made by that division.

The workhorse gear is the IBM communicating mag card typewriter, models of which are tied together with a support program called Version II of the Communications control application program/7. Data is transmitted over voice-grade lines to its destination to an IBM System/7 small computer. "The primary objective of the electronic document transfer network," says an IBM official of the gear, "is to substantially decrease the distribution time of internal correspondence."

Piggybacking. That description by IBM, of course, sounds like a definition of electronic mail. And so indeed, does the description of a new system being configured by the Digital Equipment Corp. Essentially, DEC message-switching centers in Maynard, Mass., and Switzerland will be piggybacked for electronic mail use with Decscopes and word processing gear manufactured by the firm. Some 200 DEC users, who will be able to key into the system from any terminal in the system will be on-line initially. The system will be highly interrupt-oriented because it is designed to be used within buildings and among various geographically scattered Digital Equipment facilities.

The DEC system

DEC mailbox. *Digital Equipment Corp. is piggybacking electronic mail onto its corporate message net.*



The Yankee Group's Anderson says there is "enormous interest" in electronic mail by users, who, he finds, appear to be confused by the wide range of offerings that are beginning to surface. With the thought of establishing a user-oriented information clearing house on electronic mail, the Yankee Group has been instrumental in the establishment of the U.S. Electronic Mail Association, a user-oriented group that will exchange information and experiences.

The whole problem of lack of user awareness was underscored not so long ago when managers of a Canadian timesharing network saw a national postal strike as their golden opportunity to market its electronic mail service. The firm promoted its service heavily and sat back and waited for the orders to come in. There was one response: From a woman who wanted to deliver a package to her son.

Rates drop in DDS tariff

AT&T has filed yet another tariff for its Dataphone digital service (DDS), and the tariff will become effective Oct. 23 if there are no objections. The proposed mileage rates, in most cases, are lower than the present ones, and also below the original DDS rates thrown out by the FCC early this year.

As a comparison of the different mileage rates between New York and Chicago, service with a transmission speed of 2.4 kbit/s cost $476.92 a month under the old rates; it costs $747.92 under the present rate structure, and would cost $500.24 under the proposed new rates. Other costs compare as follows: at a transmission speed of 4.8 kbit/s, the rates are $688.44 for the old, $898.92 present, and $674.64 proposed. At 9.6 kbit/s, they are $981.16 old, $1,141.92 present, and $940.24 proposed. At 56 kbit/s, they are $3,515.44 old, $4,195.60 present, and $3,300.20 proposed.

In addition to the circuit mileage, the DDS user pays for a station termination—Bell calls it a "data service unit (DSU)"—a data access line (DAL), and a hub office termination. Customers within each DDS serving area—that is, within about five miles of the hub office—lease a Type 1 DAL consisting of a baseband copper loop, while those farther out utilize a Type 2 DAL, consisting of a baseband and T1 carrier sections. The latter charges were increased considerably. For example, a Type 1 DAL operated at 2.4 kbit/s would cost $99 a month under the proposed tariff versus the current $66.95 charge. Charges for a 2.4 kbit/s DSU would rise from $15.45 to $22 a month, while the hub office termination charge would increase from $20.60 to $65.

AUGMENT TECHNICAL MEMORANDUM

COMPOSE SUBSYSTEM REFERENCE

TYMSHARE INCORPORATED
Office Automation Division
20705 Valley Green Drive
Cupertino, CA 95014
June 7, 1982

COMPOSE SUBSYSTEM REFERENCE

INTRODUCTION

The AUGMENT Compose subsystem provides the capability to output documents stored in AUGMENT files on publication-quality devices such as photocomposition units, laser printers, and Computer Output Microform (COM) recorders. It also provides facilities for proofing and verifying the page makeup prior to final publication. The design of Compose is modular, allowing easy adaptation to many types of output devices.

USING COMPOSE

Typical Steps To Use Compose

In order to publish a document using AUGMENT's COMPOSE subsystem, the document must be in an AUGMENT file. The Base subsystem of AUGMENT contains commands for inserting and editing text in AUGMENT files.

The format of the published document is controlled by ``Output Processor directives''. These are special text expressions which are inserted at the appropriate place in the document to control such aspects of document format as margins, indentations, spacing between lines and paragraphs, running headers and footers, and so on. The ``Output Processor User's Guide'' describes the usage and effects of each of the available directives. The directives for document publishing through COMPOSE are essentially the same as those for printing on workstation printers, line printers, or terminals. The main difference is that when you use COMPOSE, the ``photocomposition measure'' (a value accurate to one thousandth of an inch) will be used for the directive value, if provided. This allows the accurate positioning which is needed for most publication tasks.

When you have a document prepared with the desired format directives, the next step is to use the Compose subsystem Layout command to produce a ``layout file''. A layout file is a special type of file, not readable through normal AUGMENT or Exec commands, which contains an encoded representation of each page of the document. The layout file specifies fonts, character sizes, linework data, and exact location on the page for each textual or graphical entity on each page.

At this point, you should use one of the Compose facilities for examining the page layouts in the layout file. Often you will find mistakes in the format specification or see places where the appearance of the document can be improved. Compose provides two ways of doing this:

The Compose subsystem Proof facility requires the use of a

Genisco G-1000 or Tektronix 4014 high-resolution display. Such a terminal is typically connected to the ''printer'' connection on an AUGMENT display terminal. Text and graphics are portrayed on the high-resolution display in their correct relative positions, one full page at a time. Compose proofing commands allow displaying any page in a layout file and printing the screen image on an attached hardcopy unit. This is the fastest way to review the format of a document, and will generally be the method of choice in publishing applications of any substantial volume.

The Describe facility produces a ''description'' of the page layout of any or all of the pages in a layout file. The description is typically printed on an ordinary alphanumeric printer. It gives the full details of the fonts, character sizes, and page coordinates of each line of text, all printed out in text form. This information is more exact than the Proof display, and should be examined in any case where details of positioning are critical, or to verify font usage if you are unsure of the effects of the font directives in the document. Also, this capability allows AUGMENT users not equipped with display units suitable for Proof to carry out occasional publishing of small to medium sized documents.

When a satisfactory layout file has been produced for the document, the Compose subsystem Translate command is used to create the file that will be input to the photocomposition unit, laser printer or COM recorder. COMPOSE will determine the correct format file format to use for the output unit specified in the user's profile or in the Layout command, and the appropriate data for input to that unit will be placed in a translation file. This file can then be transferred to the output unit by appropriate means (usually by writing it on a magnetic tape).

The next three sections of this document explain the layout, review, and translation steps in more detail.

Making the Layout File -- The Layout Command

All the publishing functions of Compose work on layout files. The Layout command makes a layout file from an AUGMENT file or part of a file. The full syntax of the Layout command is:

```
Layout [(using view) VIEWSPECS] STRUCTURE (to)
File (to be named)/Append (to file) CONTENT LAYOUTDEVICE/CK
```

Certain format defaults are based on the viewspecs in the current file window. If the optional viewspecs are provided in the Layout command, they will be used instead. Explicit directives within the STRUCTURE will override corresponding viewspecs, regardless of their source.

If ``File'' is specified, a new file (or new version) is created. Specifying an existing version of an existing file is an error, even if that version is deleted (but not expunged). If ``Append'' is specified and there is not an existing undeleted file with the specified name, an error is indicated. ``Append'' may not work for some device types; ask before using it.

The layout is produced to fit the capabilities of a particular output device. For LAYOUTDEVICE, typing <OK> selects the default device from your profile. Give the command word "Device" to override the profile and specify the device.

LAYOUTDEVICE = Device DEVICE (model) MODEL (at vendor) VENDOR

For DEVICE, the devices currently supported will be available as command words. Once you have chosen a device, you may have to specify for MODEL a command word for the model of that device. If that device and model are supported at more than one vendor, you have to specify which one for the command word VENDOR.

The choices available are controlled by the ``index file''. Normally the index file will be a system file supplied with AUGMENT. In some special customer installations, the index file may be under the control of local systems managers or individual users. This is discussed under ``Configuring Compose for Special Installations''.

``Messages have been entered in log'' is displayed at command completion if a directive specified a type face or size which is not available on the device. The messages will tell what was used instead. These messages are placed in the file named NOTIFICATIONS-IDENT (where IDENT stands for the user's ident) in the login directory.

Reviewing the Layout File -- The Describe and Proof Commands

Describing a Layout

The Describe command in Compose generates a description of a layout file in a special format which tells exactly where each element of the page layout is located and what typographic specifications will be used for setting type on the output unit. You can print this description on several types of devices and you can give a wide variety of specifications to control how the description should be printed. The syntax of the Describe command is:

Describe Layout (from layout file named) CONTENT (on) OK/DEVICETYPE (specifications:) OK/DEVICESPECS

Choices for DEVICETYPE are: Workstation (printer), Terminal, Lineprinter, or File. Typing <OK> at this point selects the device specified by the Base profile feature ``print, default''.

Device type Workstation (printer) creates a workstation printer file in the Base profile specified directory for workstation printer files.

Device type Lineprinter creates a line printer file in the Base profile specified line printer directory and it is automatically printed by the line printer despooler.

Device type Terminal causes the description to be printed immediately on the terminal, which may be a display or typewriter type terminal.

Device type File creates a sequential file containing the description text.

Available DEVICESPECS for each DEVICETYPE are listed below. The meaning of Begin and End in the Describe command is different from that in the Base Print command; this is explained below. All other DEVICESPECS have the same interpretation as in the Base Print command. For all DEVICESPECS that you do not specify, you will get the default indicated in parentheses.

Workstation (printer): Begin (1), Copies (1), End (the end), Hold (don't hold), Priority (normal), Queue (nc), Wait (no)

Lineprinter: Begin (1), Copies (1), End (the end)

Terminal: Begin (1), End (the end), Formfeed (simulate with linefeeds), Wait (no)

File (to be named): Begin (1), End (the end)

The meaning of Begin and End specifications in the Describe command is slightly different from that in the Base Print command; this is explained below. All other DEVICESPECS have the same interpretation as in the Base Print command.

Begin (on page) CONTENT: The description will begin with the page that you specify for CONTENT. AUGMENT calculates where to start the description by assuming the first page in the layout file is page 1 (regardless of the page numbers in the photocomposed document or the number of pages in the printed description), counting until it reaches the page you specified. The description begins with that page.

End (on page) CONTENT: The description will end on the page that you specify for CONTENT. AUGMENT calculates where to stop the description by assuming the first page in the layout file is page 1 (regardless of the page numbers in the photocomposed document or the number of pages in the printed description), counting until it reaches the page you specified. It stops generating a description when that page ends.

Proofing a Layout

The Proof command in Compose interprets the layout file you specify and displays a visual reproduction of the pages of the layout file on a graphics display (usually attached to an AUGMENT workstation). This visual reproduction shows pagination, line breakage, the vertical placemert, and the justification of the line on the page. (asterisks on lines set flush left or flush right indicate that the full text appearing in the line cannot be fully displayed). The syntax is:

Proof Layout (file named) CONTENT OK

The first page of the layout file is displayed automatically as part of the Proof command.

If no extension is suppliec with the layout file rame, .LAYOUT will be assumed.

Typing <CTRL-O> on the workstation keyboard during page image creation causes page display creation to halt. The command prompt will be displayed and the user may give any Compose command desired.

After the Proof command is given, the following ccmmands may be used to display other pages in the layout file in any order.

Back (one page) OK

Displays the page preceding the one currently displayed.

Next (page) OK

Displays the page following the cne currently displayed.

Skip (to) Backward (number of pages) CONTENT OK

The Skip (to) Backward command in Compose displays the page that is a specified number of pages clcser to the beginning of the laycut file, counting from the current page.

Skip (to) First (page) CK

The Skip (to) First command in Ccmpose causes the first page of the current layout file to be displayed.

Skip (to) Forward (number of pages) CONTENT OK

The Skip (to) Forward command in Compose displays the page that is a specified number of pages closer to the end of the layout file, counting from the current page.

Skip (to) Last (page) OK

The Skip (to) Last command in Compose causes the last page of the current layout file to be displayed.

Skip (to) Page CONTENT OK

The Skip (to) Page command in Compose causes the page you specify to be displayed. The first page in the layout file is considered to be page 1, whether or not it bears the number 1.

If the graphics display has an attached printer, the Print command may be used to print all pages when displayed, or to print specific ones once. The profile feature "printer delay seconds" should be correctly set before using Print. Here are the forms of the Print command:

Print Every (page automatically) OK

The attached printer will print the pages currently displayed by Proof every time you change the page displayed.

Print None (automatically) OK

Undoes the effect of the Print Every (page automatically) command. It does not prevent you from using the Print Rest or Print Pages commands or the "print" button on your graphics display device.

Print Rest (of pages) OK

The current page is printed; then each succeeding page in the layout file is displayed and printed one after another, until the end of the layout file is reached. May be cancelled at any time by typing <CTRL-C>.

Pages: Print Pages (from) CONTENT (through) CONTENT OK

Beginning with the first page number specified, each page in the layout file is displayed and printed automatically, one after another, until the second page number you specify has been displayed and printed. May be cancelled at any time by typing <CTRL-C>.

Before using Proof the first time, the Compose profile features "proof, display device, type" and "proof, display device, setup" should be verified as being correct for the graphics display configuration.

Translating the Layout File -- The Translate Command

In general, each type of phototypesettter, COM recorcer, and laser printer has a unique format for input data. Furthermore, different machines of the same model belonging to different users or service bureaus (``vendors'') usually have different type sizes and styles available. The Compose subsystem Translate command takes one or more layout files and produces an input file acceptable to a particular device, model, and vendor/user. The general syntax of the two main forms of the Translate command is:

Translate Layout (file named) CONTENT (for) OUTPUTDEST (specifications:) OK/DEVICESPECS

This translates one layout file.

Translate Multiple (layout files - <OK><OK> after last) CONTENT ... CONTENT OK (for) OUTPUTDEST (specifications:) OK/DEVICESPECS

This translates all layout files as though all the pages were in a single layout file.

Available OUTPUTDEST command words are:

Vendor

Use Vendor to indicate that the file should be translated for an offline photocomposition device (that is, ore not directly connected to the computer on which you are usirg AUGMENT). The syntax is:

Translate Layout (file named) CONTENT (for) Vendor File (to be named)/Append (to file) CONTENT (specifications:) OK/DEVICESPECS

Translate Multiple (layout files - <OK><OK> after last) CONTENT (for) Vendor File (to be named)/Append (to file) CONTENT (specifications:) OK/DEVICESPECS

Available DEVICESPECS for destination Vendor are listed below. For all DEVICESPECS you do not specify, you will get the defaults indicated in parentheses.

Begin (on page) CONTENT (page 1)

The translation will begin with the page specified for CONTENT. AUGMENT calculates where to start the translation by assuming the first page ir the first layout file is page 1 (regardless of the page numbers in the photocomposed document), then courting until it reaches the page specified.

End (on page) CONTENT (last page)

The translation will end at the end of the page you specify. AUGMENT calculates where to stop the translation by assuming the first page in the layout file is page 1 (regardless of the page numbers in the photocomposed document), then counting until it reaches the page specified.

The default extension for the translated file is .TRANS.

"Translation Processor In Progress" is displayed when the translation begins successfully. When translation is complete, the message "Finished - no notifications logged" will normally appear. The message "Messages have been entered in log" is displayed instead if the layout file included some typographic specification not available on the device. The messages will tell what was used instead. These messages are placed in the file named NOTIFICATIONS-IDENT (where IDENT stands for the user's ident) in the login directory. This should happen only if the file containing font data for the device has been changed since the layout file was created. If this message occurs unexpectedly, contact TYMSHARE, the vendor, or whoever maintains the interface to that device, for information and advice.

Rasterprinter, Workstation, Terminal

The current version of AUGMENT does not support these output destinations. They will be supported in a future release of AUGMENT.

TYMSHARE provides the files and programs used by Compose to make this work with certain installations. Therefore, users must do one of the following:

Make arrangements to use a TYMSHARE supported service bureau.

Contract with TYMSHARE for full support of the device they wish to use.

Acquire the use of a supported device and provice font data needed to set up the font mapper data file.

Provide their own interface (see "Configuring Compose for Special Usage").

In any case, the final file is sent (usually on tape) to the device.

CONFIGURING COMPOSE FOR SPECIAL USAGE

Customer Developed Interfaces

Compose is designed to allow customer installations or individual users to build custom interfaces which can be accessed by normal Compose commands. Whenever Compose is to be used with a particular device installation, two modules must be provided. There must be a font mapper data (FMD) file containing the specification of available fonts, character width data, character value translation, and so forth. This will generally be different for every site, even if the same model of device is being used. There must also be a translation processor to translate a standard AUGMENT layout file into the appropriate input format for the device. This will usually be the same for different installations with the same model of output device.

In consequence, it is considerably easier to interface Compose with a device of a type already supported. All that is needed is to prepare the FMD file, which is used by the Layout, Describe, and Translate commands. The Font Mapper document listed in the References specifies the format of the FMD file. Customer provided FMD files can be added to AUGMENT at no extra cost. Special arrangements (probably at extra cost) must be made if the font data is supplied in some other form.

To interface to a device not already supported, a translation processor must also be written. A translation processor is a program of non-trivial, but not immense, scope. The documents mentioned in the References contain sufficient information for a programmer proficient in the L10 language to write one. Nevertheless, customers intending to do this are advised to contact the AUGMENT Development Unit before beginning.

Linking Interfaces with Compose

The connection between device, model, and vendor names and FMD files and translation processors is established by an AUGMENT file called the ``index file.'' The system default index file is located in the release directory (normally UREL) and is called COM-DMV-DATA.AUG. Beginning with AUGMENT version 10.14, a capability exists for a user to specify a different index file via a Compose subsystem profile feature. The index file is structured in the following way:

The name of each legal device type is placed in a first level statement.

The name of each model of each type of device is placed in a second level statement below the device. This must be done even if there is only one legal value for the ``model''.

The name of each vendor (or user) of a given device and model must be placed in a third level statement beneath the device and model.

Under each third level (vendor) statement is a plex at level 4 containing the interface module filenames in operating system format. At present, two such modules are defined; thus two filenames must be provided. If no directory is specified, the current release directory will be assumed at execution time.

The first fourth level statement must contain the filename of the appropriate translation processor program.

The second fourth level statement must contain the filename of the appropriate FMD file.

All device, model, and vendor names should consist only of printable characters. There should be no ``stray'' characters anywhere in the file (except for the origin statement, which may contain anything at all).

The TYMSHARE-supplied FMD files and translation processors are located in the AUGMENT release directory. Sources for each are in the directory OPSRC on those systems where sources are supplied.

COMPOSE PROGRAMMING REFERENCES

The following documents contain the additional design information needed to program an interface between COMPOSE and an output device.

Layout File Format <progdocs, layout-file, :wh>

Describes the internal format of the layout file. This is necessary information for programming a translation process.

Font Mapper <progdocs, font-mapper, :wh>

Describes the Font Mapper software and defines the format of the font mapper data file. In general, a font mapper data file is needed for each device with which COMPOSE is to be used. This document also contains information about how to set up font data specifications that can be processed by the FONTLCAD subsystem to produce the font mapper data file.

Translation Processor Interface <progdocs, translation, :wh>

Describes the interface protocol which must be employed to allow a translation processor to be used by the COMPOSE Translate command.

AUGMENT TECHNICAL MEMORANDUM

NEW IDENT SYSTEM

TYMSHARE INCORPORATED
Office Automation Division
20705 Valley Green Drive
Cupertino, CA 95014
October 6 1981

NEW IDENT SYSTEM

General Design

Requirements (background)

Meet all Ident needs of the new MAIL subsystem.

Multiple Ident Systems

This design supports multiple ident systems, each system identifiable by name. To specify an individual by icent, you also need to specify the ident system name, if different from the one in which you are currently running.

Ident uniqueness

At any given time, one ident will represent one individual within a given ident system, but not necessarily across ident systems. The same ident in different ident systems may represent the same or different individuals. One individual may have different idents in different ident systems.

Multiple Hosts

Across hosts

Multi-host system

A single ident system can span several hosts. A single host can support several icent systems.

Changes

A change made to the data base of one host in a multi-host ident system will be propagated to all hosts in the system. As much of this process as is feasible will be done automatically.

Support

"Hosts A and B support Ident system X" means that there are database info files with lookup facilities for Ident system X on hosts A and B. If host C does not support X, then there will be facilities for C to interrogate X on A and B and have the relevant information returned to C.

Mail addressing

By addressing a system X ident, a user on host C can send mail directly to a user associated with ident system X; i.e. without querying host (A or B).

Protections

Between Ident Systems

Each ident system will restrict access to its idert databases and journals. This means that if ident system Y wants to exclude ident system X from its database then X users cannot, in general, query Y. An exception to this rule is made for MAIL: if the X user knows a Y ident, he may send mail tc it.

Journal Access

Each journal facility will have the ability to restrict read/write permission to a specific group of individuals. The journal facility will use the ident system as one method fcr restricting access.

Active and Inactive Idents

There will be an inactive database for each ident system which will contain all expired idents. This will be used to cetermine whc was associated with an ident at a given time and date, allowing us to reuse inactive idents. At a particular time within an ident system, no two idents can be the same.

HS FILES, Ident Files

Since certain ident system information must be accessed very quickly - information used by the mailer, for example - all information where the access-time is critical will use the HS file system.

Icent Lookup

The BE ident support procedures have to be changed tc the HS file procedures to find an ident record.

Login Name Lookup

The getuid procedure in the FE has to be changed to use the new lookup by login name process. Does the BE look up the ident by login name as well?? It shoulc get the ident from the FE.

Database Structure

Active Idents

The active database will ccrtain information relating to idents currently "in force".

New Ident Types

There will be two new types of idents, role anc program idents. Role idents will include such things as FEEDBACK and OPERATOR.

Inactive Idents

The inactive idents database will look similar to the active database Inactive ident records will include the expiration date.

Ident Systems

There will be an inverted table that contains each ident systems home host and the hosts the system is supported or. However this information can be accessed by the ident subsystem.

Idents as AUGMENT Signatures

Limit of 4 letter/digits.

Ident Resolution

Over Ident Systems

There will be a way to find out which ident system a given signature in a given file is definec in. Such info need not be printed with each ident, but will be available via some facility.

Over Individuals

There will be a way to find the individual from the statement signature. The date anc time in the signature will be resolved to the individual who had that ident cn the given date.

Compatibility.

The new ident system will not make old files or their idents obsolete in any way.

Larger Ident Systems

Cne ident system will accept as many idents as reasorably possible.

New Ident Subsystem

Database modification.

It will be possible for a person or persons to adc new idents, delete old idents, and change user information associated with an ident.

Easy to use.

The maintenance system will be easy to use and will do automatic verification, etc. Use may be restricted to administrative personell only, but not necessarily system personell.

Insert.

It will be possible to formulate and insert a new ident. It will be required that the new ident be completely specified before any entry is made. Any necessary corrections to existing idents will be made automatically. The formulated ident information will be maintained while updating. If errors exist the user will have the ability to edit the entry and try updating again.

Groups.

If the new entity is a group, records of all group members will be acjusted automatically to show group membership.

Organizations.

If the new entity is an organization, records of all organization members will be adjusted automatically to show organization membership.

Individuals.

If the new indivicual is in a group and/or organization, those entities will be charged to show that they include the individual.

Delete.

It will be possible to delete idents from the system. Automatic correction of other ident information will be made. It is acceptable to run a utility after many deletes are made, in order to maintain efficiency of information.

Change.

It will be possible to change ident information. Changes will be edited using AUGMENT. Any updates to existing idents will be made automatically.

Login Creation

The ident subsystem will eventually reach through to the TENEX subsystems that create a login directory and do everything else that has to be done to add a new user to the system. This may be postponed until the major part of the ident system is complete.

Program Interface

General Database Management System

The database management system will be as general as possible so that it can be used on other AUGMENT files as necessary. The capability off adding, deleting and changing the database will be general. Record accessing will also be general. A record

structure will be created for any database and the program interface will use the record definition for these general capabilities.

Soundex Name Lookup

The Soundex method of hashing will be used for names when the exact name is not known (misspelled, mistyped or not know by user exactly). This method will pick up similar names and the user can pick the correct one or discard the help. The present method of entering partial names will no longer be available.

MAIL Interface

There will be information available at login time for MAIL. This will consist of the directory of the login users mailbox if they recieve mail on the logged in host or the host which they do recieve mail on. In addition all the information for mailing to anyone else in any ident system on any host will be accessible.

New User Subsystem

Finding Correct Spelling of Names

As described above the Soundex method of name resolution will be used to help find the person to which the user wishes to send mail even if the spelling of the name is not known.

Finding Idents from Names

There will be a facility to find a users ident given their last name.

Finding Information from Idents

There will be a facility to get all or a portion of the information stored in the ident database for a particular user, group, organization etc.

Finding Ident System Info

There will be a facility to find which host an ident system lives on and which hosts it is supported on.

Login Interface

The ident system provides a facility to get from a user login name on a given host, to an ident that is permitted under that login. Other mailing information will be available to make mailing faster.

Login name lookup.

The lookup given implied host name and login name, will return ident, and will be as fast as possible (used at ALGMENT entry

time). The other mail information will be the mailing directory if on this host or host of the mailbox.

Multiple ident systems.

Always returns ident system name as well, with each ident. Only one such lookup facility per host.

Network Interface

Ident Server

Ident System Creation

It will be possible to create a new ident system (that is supported over a given set of hosts). This may require help from system personnel. It must not require any program changes.

Creation empty.

It will be possible to create an ident system with no entries.

Knowledge of Existence

In general, hosts will be required to know of the existence of ident systems that are not supported on that host, and also know the "home" host for that ident system.

Ident system directory.

There will be a database that is available to both programs and human users, that contains the names of known ident systems and associated home host names and/or addresses necessary for forwarding mail and making enquiries.

Directory update scheme.

There will be a capability (which is not necessarily automated) to update the ident system directories as ident systems and supporting hosts come and go.

Functional Specifications

Beginning the Modification run

General

To modify the Ident Database, enter the Ident subsystem and execute a Begin Modification (run) command. The system will prompt you for the name of the Ident Database, the name of the branch to contain new/modified records (usually "Modifications"), the name of the branch to contain error messages, and the name of the branch tc contain the audit information (usually "Audit-trail"). These branches are in an AUGMENT history file called <Ident-system-name>.AUDIT. AUGMENT opens the .AUDIT file, and is ready for the other commands (Add/Modify/, etc)...

Making new AUGMENT records for later inclusion

General

For each "Add" command, the Ident Subsystem builds a "record", an AUGMENT branch, derived partly from information with the command itself, and, in the case of information that is likely to remain constant for entries within a given organizatior (such as mail type, or host), the "proposed fields" are copied from another record in the same organization. Although, typically there is much duplication for records within an organization, any particular field may differ and can be edited after the system has built the record. Base texit-editing facilities are available within the Ident Subsystem; they can be used to modify any record produced by the system.

Add Individual (Record):

AUGMENT prompts you:

for full name

for telephone number

to ask if ident it has just formed is acceptable

for an ident of an individual similar to the the one tc

be added, or the name of his organization

From the information you have just entered and the fields copied from the record used as a model (same organization/similar individual), the Ident subsystem makes an AUGMENT record. Nothing placed in .AUDIT file at this point. Branch will look like the following:

```
KMD: NEW INDIVIDUAL
   NAME:  DICNNE, KATHLEEN M.
   Addr:  TYMSHARE Inc.
          20705 Valley Green Dr.
          Cupertino, CA 95014
   Tele:  (408) 445-6643
   Org :  IBM

   Host:  DICNNE@Office-5 DICNNE@Office-5
   MAIL:  KDICNNE@OFFICE-2 KDICNNE@OFFICE-5
   Type:  ARPA
   Grps:  TYMSHARE, WOMEN
   Date:  1-Sep-81
```

(After the AUGMENT record is made, you may edit it, if necessary. You can follow the "Add" immediately by an Process Modifications command, or you can make a series of new records and modifications and enter them into the Ident Database together. (See discussion under "Modification of Ident Database Records").

Add Group (Record):

Ident subsystem prompts you for name of group and initializes the "memb" (members) field to NULL. "Add Individual" commands whose "grps" fields reference this group will cause the Ident System to add their names to the "memb" field of the appropriate group record, automatically. However, it may be more convenient, when you create a new group and the individual records already exist, to add members directly to the members field, and the Ident subsystem will make the required changes in the Individual records You can do this by replacing the "memb" field by any number of idents in the form: (*+ / *-)ident, (*+ / *-)ident, ..., . For any ident preceded by a plus-sign, the ident system will add the entry to the "memb" field and to the "grps" field of the individual record. Similarly, entries preceded by minus-signs will be deleted from both records. To summarize, additions and deletions of idents can be made by entering +ident or -ident in the Group record OR the Individual record and the Idents system will make the other change for you. The new record will look as follows:

```
DARCOM-GENERAL: NEW GROUP
   Psid:  DAR01
   Coor:  DIA
   Memb:
   Date:  1-Sep-81
```

(See discussion at end of Create (AUGMENT) Record for Individual).

Note on Psid field: At the same time the ident system processes the Create Record (for) Group statement, it creates an additional record, a Psid (pseudo-ident) record. Group names are, in general, too long to fit into one word, so groups are issued ident-like abbreviations which consist of the first three letters of the group name plus a two-digit sequence number and are used to identify the group when referenced in HS file fields. These pseudo-idents are translated back to names when group fields are displayed to the user. They can be distinguished from idents by their five characters, one character longer than the longest legal ident.

Add Role (Record):

(Similar to Add Individual).

The Ident subsystem makes an AUGMENT record like the following:

```
FEED: NEW ROLL
   Name:  FEEDBACK
   Auth:  PAMV, SKB2
   Addr:  TYMSHARE Inc.
          20705 Valley Green Dr.
          Cupertino, CA 95014
   Tele:  (408) 445-6643
   Org :  OAD

   Host:  FEEDBACK@Office-5
   Mail:  FEEDBACK@Office-2 FEEDBACK@Office-5
   Type:  ARPA
   Grps:  TYMSHARE, WOMEN
   Date:  1-Sep-81
```

(Editing and entry into the database as in other "Add" commands).

Add Organization (Record):

This command creates a NULL organization in a manner similar to the Add Group command. It prompts for organization "Ident", full name, and coordinator and creates a record like the following:

```
OAD: NEW ORGANIZATION
   Name:  Office Automation Division
   Coor:  WJE
   Addr:  TYMSHARE Inc.
          20705 Valley Green Dr.
          Cupertino, CA 95014

   Tele:  (408) 445-6643
   Mail:  EISMANN@Office-2
   Type:  ARPA
   Date:  1-Sep-81
```

(Editing and entry into the database as in other "Add" commands)

Add Program (Record):

(Similar to other Add Record commands).

Ident will build a record like the following:

```
DBMS: NEW-PROGRAM
   Name:  Data Base Management System
   Coor:  JDH
   Host:  DBMS@Office-5 DBMS@Office-2

   Mail:  DBMS@Office-2
   Type:  ARPA
   Date:  1-Sep-81
```

(Editing and entry into the system as in other "Acd" Commands).

Modifying existins Ident System records

General

The Ident subsystem has BASE text editing facilities. To change information withing the Ident System, execute a Modify command specifying record type and name/ident and the fields that you need to modify. The system will make an Augment copy of the record you have specified which you can edit, and subsequently reenter.

Modify (Individual/Group/Role/Program/Organization) (Record) command

Makes AUGMENT copies of all fields of all Ident Database records referenced by this command. When creating copies of individual

or role records, you may use the name of a group or organization to stand for "all records of individuals in this group/organization".

Whereas field contents can be modified, statement-structure should be left as is:

each field is a statement

an occasional NULL statement is included as an aid to the eye

some fields, like the name field, contain multiple lines,

separated by EOL's

statements have name-delimiters of NULL, colon

statements have 3-4 character names

field contents is everything following the colon

For most fields, the field in the AUGMENT copy replaces the corresponding field in the Ident Database record. The Ident Subsystem examines certain AUGMENT copy fields, however, for editing directives as follows:

+ followed by a visible

means append this visible to corresponding field in Ident

Database record

- followed by a visible

means delete this visible from corresponding field in Ident

Database record

The following fields use the +/- editing directives:

"Grps" field (in individual record)

"Memb" field (in Group Record)

"Host" field

With a display you mark fields to be edited; with a TI, you address statements by Ident and field names, KMD!addr for example. Ordinarily, there is only one statement with a given field name within one record. However, in certain cases the text has exceeded the string capacity of an AUGMENT statement and therefore a second, unlabeled statement, is added, down a level from the first.

(see the Enter Modifications command for how to get these records back into the system).

Terminating Records

General

The Terminate command, makes an AUGMENT termination record (with a proposed termination date) and enters it in the Modification branch. It will cause the current Ident Database version of the record to be moved to the inactive file after verification is complete.

Syntax

Terminate Individual

Terminate Group

Terminate Role

Terminate Organization

Terminate Program

Deleting Records

General

The Delete command, is usec in cases where an errcr has been detected, the record shoulcn't have existed. In cases where the exist was valid, use the Terminate command.

Syntax

Delete Individual

Delete Group

Delete Role

Delete Organization

Delete Program

Processing edited AUGMENT records into the Icent Database:

Process Modifications

This command accepts all records in the "Modifications" branch of the .Audit file: those produced by the "Add", "Delete", "Terminate" or "Modify" commands or records that have the same format as any of these.

Description of AUGMENT record Input to "Process Modifications" command

New record fields are described in the "Add" commands. Headers produced by these commands are significant for the "Process Modifications" command and so are documented below:

New record headers are as follows:

<ident> "NEW" ("INDIVIDUAL" / "GROUP" / "ROLL" /

"ORGANIZATION" / "PROGRAM")

For example,

```
KKD: NEW INDIVIDUAL
   Name:  Dionne, Kathleen K.
   Addr:  TYMSHARE Corporation
          20805 Valley Green Dr.
          Cupertino, CA 95014
   Tele:  (408) 445-6643

   Orgn:  IBM
   Host:  DIONNE@Office-2
   Mail:  KDIONNE@Office-2
   Type:  ARPA
   Grps:  TYMSHARE, WOMEN
   Date:  1-Sep-81
```

Headers made by Modify commands requesting replacement of all fields are as follows,

<ident> "FULL" ("INDIVIDUAL" / "GROUP" / "ROLL"

/ "ORGANIZATION" / "PROGRAM") "MODIFICATION"

Example of whole replacement of "group" record,

```
NPG: FULL GROUP MODIFICATION
   c#:  DIA
   m:  +PDG, -XYZ
   d:  1-Sep-81
```

For partially-modified records (Only certain fields were designated on the "Modify" command) we have headers as follows,

<ident> " "PARTIAL" ("INDIVIDUAL" / "GROUP" / "ROLL" / "ORGANIZATION" / "PROGRAM") "MODIFICATION"

Example of partial replacement of "group" record (only adds member "ABC" to the group),

```
NPG: PARTIAL GROUP MODIFICATION
   Memb:  +ABC
```

Verify Modifications

This command informs the Ident Subsystem that all entries have been made so that cross-checking and other verificatior can procece

Keeping an Audit Trail

General

The Ident subsystem provides an audit-trail with a record of exactly what has been done since the last update. More specifically, it creates a branch in ident-name.icent with every AUGMENT structure accepted by the Enter Modifications command. Any record that changes state by this run will be copied under the the audit-trail branch, including a copy of the way it was before the change(s) were made.

In cases where disk-failure, system crash or other disaster has caused the HS file to go bad while processing modifications, it may be possible to recover by using the ident-name.audit file. Since all operations on the ident file are ones that, either cannot be done twice (for example, deletions), cr have the same result if done twice (for example, replacements), it doesn't really matter at what point the update failed (as long as we have the ident database as it was before the update plus the ident-name.ident file). The Reprocess Modificaticns commard should be used in this type of system failure.

Reprocess Modifications commanc

Syntax:

Reprocess Modifications

You use this command when system or hardware failure caused the updated HS Ident file to go bad. The Ident subsystem reads the .AUDIT file to determine which modifications you entered and reenters them for you, allowing you to add any adcitional

modifications you may need at the end. Entries that don't crosscheck or contain other errors are referenced in the errors branch.

Ident system files

<ident-system-name>.ident

The main file in the Ident System Database

<ident-system-name>.audit

Contains recent history of adcitions and changes to the Database

<ident-system-name>.mail

Gives idents under a given <login-directory>@<host-name>

<ident-system-name>.last

Gives idents under a given last name

<ident-system-name>.sndx

Gives idents with a given SOUNDEX code; i.e. idents of people that have names that sound alike are related

AUGMENT TECHNICAL MEMORANDUM

Foreign Mail interface

TYMSHARE INCORPORATED
Office Automation Division
20705 Valley Green Drive
Cupertino, CA 95014
June 15 1981

INTRODUCTION

The following is a sketch of a design for interfacing to the Augment mail system Foreign mail systems.

The Arpanet mail system is dealt with in particular.

SPECIFICATION

For the Arpanet Foreign Mail interface

Requirements

Place Arpanet mail into the Augment Mail system

Identifiers

Arpanet mail addresses must be translated into idents for those receiving Augment Mail.

Format

As much as possible, paragraphs in sequential text files should be translated into paragraphs and extraneous carriage returns should be elimitated.

There should be some way for the reciever to specify how material is to be translated.

No information should be lost in the translation process.

Send Augment Mail out to the Arpanet mail system

Identifiers

Arpanet addresses in a distribution list must be recognized as such.

Arpanet recipients must have their idents translated into valid Arpanet mail addresses.

Augment mail users' idents going out to Arpanet mail recipients must be translated into a valid Arpanet mail addresses.

Format

The sender should have the ability to specify how the Augment structure will be translated into sequential form.

Functional Specs

Coming in

Identifiers

Non-Augment addresses that do not have an at-sign in them will have an at-sign appended to them.

Format

A field in the ident system "Augmentize files using:" will allow you to specify how you want sequential files Augmentized.

One

If you want to make sure that the original message is all there and the heuristic algorithm does you no "favors" you set this field to "One <CR> between statements" in which case each line of the message is made a separate statement preserving invisible characters.

Two <CR> between statements

Assembler

Heuristic

All flavors of Augmentization will attempt to structure messages containing messages with each sub-message in a branch.

Going out

Identifiers

Arpanet addresses will be recognized in a distribution list by having an at sign in them.

Arpanet recipient idents will have fields in the ident system to indicate their valid Arpanet address, if they wish to recieve their mail in that manner.

Augment recipients' idents will be translated into the following form for items going out to the Arpanet: IDENT!IDSYS where IDENT is the IDENT of the Augment user. IDSYS is the name of the Ident system in which the IDENT is valid.

Format

The default

If nothing special is specified, the sequential file will be created via the Create Sequential commanc code or the structure specified using viewspecs: wljnpyCFGILPQ

The sender specifies different viewspecs and whether the body goes through the Print Formatted coce via a field in the header

Formatted: Yes/No VIEWSPECS

[This is likely to trip people up since you have to specify whether or not you want to use the CP in order to specify a viewspecs. I think we should have a separate Viewspecs field. -- kirk]

DESIGN

For the Arpanet Foreign Mail interface

There will be four stages of development.

Stage 1) runs on office-5 only uses a kludge ID system.

Stage 2) maybe office-4 and 5. FTP forwarding data base knows whether or not each user wants Augment mail.

Stage 3) Ident system knows whether or not each user wants Augment mai.

Stage 4) SMTP (Simple Mail Transfer Protocol) implemented.

Coming in to Augment

A background system job has the following modules:

directory reader module

LOOP

Reads [--UNSENT-MAIL--].PERSON&HOST file names from the directory <GATEWAYMAIL> containing items delivered to this host

Teleslave is fixed to not poke mailer.

If HOST is the current host % will this always be true? %

AND PERSON wants augment mail (send to the idert getter)

```
IF augusr($person: result) THEN
   BEGIN
   % Create Augment file - %
      firstime _ 0;
      % open augment file %
         stid _ origin;
         stid.stfile _ ccrefil( from-arpanet-mail.SOON
            Keeps the file open the whole time
               fl.flrcclos _ TRUE;
```

```
            hderstid _ agmtzr(stid, levdown, infile, idlist,
            augconv);

         creamf($person, $hderstid);

         END

      ELSE

         BEGIN

         IF person.L THEN % send as arpanet to address in person
         %

         ELSE % rename to
         <OUTMAIL>[--unsent-mail--].USERNAME^V@HOST %

(augusr) % ident getter module %
PROCEDURE(person REF % => TRUE/FALSE,augconv %);

   % updates person with IDENT/ADDRESS/FALSE %

   If PERSON

      has a dot "." in it then

         If right of the point is a valid identsystem name then

            If left of the point is a valid ident in that ident
            system then

               If that ident wants Augment mail

                  Reformat as IDENT(IDSYS).

                  Create a distribution list containing this ident.

                  Return.

         Return false.

      is a directory for an ident system known to this host then

         If PERSON is in more than one ident system then

            Either

               Return the message to sender and generate a message
               to each ambiguous PERSON informing them of the
               attempt to mail and the need to tell potential
               correspondents to use their IDENT!IDSYS or
               USERNAME@THEIRHOST@GATEWAY.

            OR
```

Anyone without a dir on GATEWAY must always have the address USERNAME@THEIRHOST@GATEWAY then send only to username on gateway host.

Else

IF that ident wants Augment mail

Create a distribution list containing this ident.

Tell the Augment structure creator what create Augment algorithm this IDENT wants.

Return(TRUE).

ELSE

BEGIN

RETURN(FALSE);

END

% if we get here, its not to be converted %

Stage 1)

Rename file to [--unsent-mail--].USERNAME ard poke mailer.

```
(agmtzr)  % Augmentizer %
PROCEDURE(stid, levadj, augconv % => hdrstid %);
```

% Creates the Augment structure from the body according to the wishes of the receiver as specified in their ident system.

Creates the header assuming the Arpanet standard.

Changes "at" to an at-sign. Appends an at-sign on the end of those that dont have one.

Calls creamf

```
(creamf) % Create Augment Mail File (.PRICRTY creator module) %
PROCEDURE ...
```

Creates a collapsed file if necessary

Or rename to FROM-ARPANET-MAIL

with extension .SOON / .RUSH / .DFRD containing the augment structure.

Its created in the <IN-MAIL> directory to be picked up by the Augment mailer.

Appends the distribution list (one ident) to the file.

design discussion

11 Dec 1981 0554-PST Barns: Re: Mailers, Mailboxes, etc.
Distribution: ANDREWS, KELLEY, DAUL, KLEISER, barns
Received at: 11-Dec-81 05:54:00-PST

In response to the message sent 10 Dec 1981 1728-PST from Andrews

I think Phil will go along with us coing SMTP mailer, judging by the tone of his msg (the one about MSG being trashy). Do not think Ellen has ever worked on Mailer...

I definitely prefer the dia!oad&office style of addressing, not so much because I particularly like idents, but it gives us more flexibility for making mailboxes wherever we want. Right implementation should even allow multiple message.txt mailboxes in one directory - with distinct names of course. Requires putting mailbox file name in some HS file, but this is not a bad idea anyway, as it will make the message.txt/mail.txt question work cut easily.

We better make sure that Allan is informed of whatever fields and things need to exist in ident database to make this happen...

I hope that a Role ident can be of more than four chars length. This allows a right solution to mailing things to offices... One more thing. I think we should make a point of using the (name) construct of RFC733 etc., for Arpa mail. DIA!OAD is pretty weird looking as a sender, would be better to expand it with a name in parens. Same thing with Role idents as sender/from/etc. Then we can handle all the govt office symbols and expand them to reasonable things- for example,

From: OSS!NASA (NASA Office of Space Science)

One other thing. WATCH OUT for the Arpa mail A@B@C addressing construct. Any number of @host thingys are legal and only the last one

needs to be a valid ARPA host. The current mailer we have does not
understand this correctly, and as a result you cannot answer a msg
from blapo@mit-eecs@mit-ai. Whatever code we write that looks for
atsigns, should probably look for atsign followed by a valid ARPA host
name (or number?) I guess you really need to scan from the end
forwards to eliminate all ambiguity because the address
blapo@100@200@300 is legal. Probably should
disallow gateway host lookup if address ends with valic hostnumber.
-bill

11 Dec 1981 0949-PST Andrews: Re: Mailers, Mailboxes, etc.
Distribution: BARNS, KELLEY, DAUL, KLEISER, andrews
Received at: 11-Dec-81 09:50:09-PST

In response to the message sent 11 Dec 1981 0553-PST from Barns

Yes, I think a role ident can be a long one (right Allan?). this implies that a person filling a role doesnot enter AUG as the
role-ident but as the person filling the role, which is what most people
want anyway...
The (name) thing is nice and needs to be done by the Send code in MAIL
when the new ident system comes up. THis is only done for the From, Sender, Author fields right?
--Don

7 Jan 1982 1655-PST Andrews: Re: arpaug and message.txt file
Distribution: KELLEY, daul, lieberman, frerch, barns, andrews
Received at: 7-Jan-82 16:57:45-PST

In response to your message sent 7 Jan 1982 1513-PST

Sounds like it will work but I would rather avoid changing tenex mailer.
Seems simpler to not run mailer autojob on hosts where you are doing ARPA-AUG
conversion and run mailer as inferior fork when you need to do local delivery
ARPANET style...But then, you are doing it.

Alternate interim ugly hack: Change the mail forwarding database on the
host doing ARPA-AUG conversion to that users that get AUG mail are listed
as receiving mail on some wierd host-- so it gets queued for

tymail as
[--tymnet-mail--]. Then find a way to grab it from tymail and deliver
it as AUG mail. This will work for queued of course. Immediate will
never work anyway since we have no ARPANET link. Use a "wierd"
hostname so you can identify them... --Don

7 Jan 1982 2002-PST Barns: Re: arpaug and message.txt file
Distribution: ANDREWS, KELLEY, daul, lieberman, french, barns
Received at: 7-Jan-82 20:03:10-PST

In response to the message sent 7 Jan 1982 1655-PST from Andrews

It sounds to me like anything that involves changing tenex mailer or
mailforwarder databases is more of a pain than anything that doesn't.
If that's true, it seems to argue for mailer as an inferior of the
conversion process. The conversion guy can look at mailer.flags and
catch the queued stuff. We could avoid the trouble with immediate
sends by just patching the sndmsg or those hosts to queue even if
you don't say to. On an Arpa host, life is tougher. Don't see
an easy answer. Probably just wait for SMTP? Else
raid Gateway-mail. ugh. -bill

8 Jan 1982 1447-PST Kelley: arpaug and message.txt
Distribution: ANDREWS, BARNS, daul, lieberman, french, kelley
Received at: 8-Jan-82 14:48:25-PST

You guys seem to think it will be easy for arpaug to go around to all directories
and find queued mail? More likely, I dont see the whole algorithm involving
running mailer as a subfork.

8 Jan 1982 1524-PST Andrews: Re: arpaug and message.txt
Distribution: KELLEY, andrews
Received at: 8-Jan-82 15:24:29-PST

In response to your message sent 8 Jan 1982 1447-PST

Kirk, all you have to do to find queued mail is periodically look in
<system>mailer.flags for bits that are on-- anc the code exists in our

Mailer already, only works on a different file. When you find a
bit on you do a gtjfn/gnjfn for files with the right name. I think starting mailer as a subfork is easy, probably a startup location
such that it will not interact with tty, or maybe if detached...

What you do is collect unsetm-mail files that need to be delivered
on this host in a dir, then run miler when connected to it. It will
do the local delivery for you. I will probably also move files to
<gateway-mail> for you if the extention includes the @host. problems I don't see? --Don

Going out to the Arpanet

Send command ident distribution verification

If ambiguous, Arpanet addresses must contain an at-sign or else they are called illegal idents.

An Augment Mail file is created having appendec tc it an "envelope" containing: The undelivered distribution list.

Augment Mailer module

To minimize the number of files that must be transported via Tymnet to the Arpanet gateway, the Going out format conversion code only runs on Arpanet gateway hosts.

On originating host

Move the Augment Mail file to an Arpanet gateway host. Strip from the undelivered distribution list those acdresses that have forked off along the way.

On gateway host

Convert all idents that want to receive mail in the Arpanet format to their Arpanet address.

Place in the <IN-MAIL> directory, a copy of the Augment Mail file containing the undelivered dist list of the Arpanet addresses of

those IDENTS that wish to receive via Arpanet and

those visibles in the original dist list cortaining an at-sign.

directory reader module

Look for files in the <IN-MAIL> directory.

Decompress the file if necessary

Strip off the appended dist list "envelope" and give to the header maker.

Give them to the sequentializer.

header maker module

Read properties

Convert them to Arpa standard fields.

Add to each ident, that doesnt already have an at-sign in it, its @GATEWAYHOST name.

If the IDENT's SYSTEM's home host is a gateway for that mail system, then use it

else use the local GATEWAY

Determine how to format the body and tell the sequentializer module.

sequentializer module

Creates a sequential file according to the format specified in the header.

Reads the dist list from the appended "envelope".

For each ADDRESS in the dist list

copies to a file named <IN-MAIL>[--UNSENTMAIL--].ADDRESS

Pokes the Arpanet mailer.

For the Arpanet Foreign Mail interface

Summary

initial design began: <date>

design review:

initial coding began:

code review:

testing began:

in experimental release:

field test release:

utility release:

Design Review Notes

Code Review Notes

Changes History

each branch here is a discussion of a "major" change in original design. when done, this change should be incorperated in all appropriate sections of this document as well as here.

Features/Changes for Future Project

each branch here is a topic that has been chosen to be NOT implemented in this project, but is a candidate for future work in this area.

BUGS AND SUGGESTIONS

For the Arpanet Foreign Mail interface

AUGMENT/NLS MAIL

JAKE, 26-Jun-79 10:24 < FEINLER, NLS-MSG-DELIVERY-MECHANISM.NLS.1, > 1

< MAILDEV, DELIVERY-MECHANISM.NLS;6, >, 23-May-79 16:03 DIA ;;;;

Introduction

This is a description of the mail delivery mechanisms that will be used for the MAIL subsystem. The mechanimsn that are not needed for tne EOP benchmarks are identified.

This is a tree-structure analysis of the different cases and what steps are necessary in each case. Many cases are duplicates.

Mail Sending

To Journal

Mechanism:

Delivery of an item to the journal for recording will be done the same as far as the SEND function of MAIL is concerned in all these cases: executing on journal master host or non-master host; immediate or deferred delivery; long or short message. The outline is left in this branch just to show the different cases that are possible to distinguish.

The MAIL SEND function will create an AUGMENT file in a directory with public "create-file" access and designated to receive new journal items for the journal in question (i.e. there will ɔe one such directory for each journal on a host). This file will contain the journal item with the header fields included in some fashion (text and/or properites, as convenient), and will include any AJGMENT properites that may have been in the original. The name of this file will be the journal number. It will be entered in the journal catalog by a background process that will periodically check this directory for files to ɔe entered.

If this is done on the journal master host, the background process will FTP the item to all other hosts which support this journal. These will go into a directory in a similar fashion and be entered in the catalog on each host by a background process on each host. However, these processes will be able to distinguish the files entered by MAIL from the files entered by another catalog-entry program (in order to avoid sending a copy of a file right back to the master host that just FTPed the file). The method of destinguishing will be determined later. If the host is not the journal master host, the background process will FTP one copy to the master host, and it will be identifiable as to which host it came from. The master background process will then FTP copies to all other appropriate hosts (not the one it came from).

Same Host (i.e. executing on journal master host)

Immediate Delivery

Deferred Delivery

Diff Host (not for benchmarks)

Immediate Delivery

Deferred Delivery

To MAIL Users

Same Host

Immediate Delivery

Unrecorded Mail (not for benchmarks?)

Mechanism: Append message header AND BODY to appropriate message.txt files, or create unsent mail files if cannot open message.txt. Signatures and non-text properties are omitted.

This is the case where we may want to warn the user if the ARE properties such as graphics.

Recorded Mail

"Short" messages (not for benchmarks, i.e. parameter = 0)

Mechanism: Append message header AND BODY to appropriate message.txt files, or create unsent mail files if cannot open message.txt. Signatures and non-text properties are omitted.

Note: signatures and properties preserved in the journal.

"Long" messages

Mechanism: Append message header WITHOUT BODY to appropriate message.txt files, or create unsent mail files if cannot open message.txt.

Deferred Delivery (not for benchmarks)

Unrecorded Mail

Mechanism: Create unsent mail file ([--UNSENT-MAIL--].name) from message header AND BODY. Signatures and non-text properties are omitted. This and other "unsent mail" files will be delivered by existing Mailer, except for cases were it is necessary to go across Tymnet to deliver it (this will not be done until we write our own mailer).

This is the case where we may want to warn the user if the ARE properties such as graphics.

Recorded Mail

"Short" messages

Mechanism: Create unsent mail file ([--UNSENT-MAIL--].name) from message header AND BODY. Signatures and non-text properties are omitted.

Note: signatures and properties preserved in the journal.

"Long" messages

Mechanism: Create unsent mail file ([--UNSENT-MAIL--].name) from message header WITHOUT BODY. Signatures and non-text properties are omitted.

Diff Host (not for benchmarks)

Immediate Delivery

same as same host, immediate, except that instead of appending to a message.txt file: start a sub-fork process that will attempt to perform cross-net delivery via FTP or an ARC-written Tymnet file transfer--this will be the ARPANET mail protocol (appends to message.txt on TENEX) or a similar thing over Tymnet.

Deferred Delivery

same as same host, deferred

Note that current Mailer will get the unsent mail files over ARPANET, but we need to modify it or write a parallel mailer to get over Tymnet (some hosts will need only one of these, some host will need both).

To Non-MAIL Users (not for benchmarks)

This is the same as the "short" message cases for same/diff hosts --i.e. the message body is always included.

Mail Reading

Note: mail from same/diff hosts, MAIL and non-mail users, immediate/deferred, all arrives in the same fashion-- in the message.txt file. "Short" messages appear in full, "long" is just citation (message header fields).

The process that reads the message.txt file and inserts info in the new mail branch can construct fairly accurate signatures on each item from the From (or Sender if present) field and the Date field (to get ident and date/time).

Do Not publish or distribute Augment/NLS Mail

JAKE, 26-Jun-79 10:19 < FEINLER, NLS-MSG-COMMANDS.NLS.1, > 1

< MAILDEV, COMMANDS.NLS;20, >, 30-May-79 15:00 DIA ;;;;

INTRODUCTION: Syntax of MAIL Subsystem commands as proposed by Syntax Committee

Note that we have started with commands needed for EOP benchmarks.

GENERAL COMMENTS: Place general comments below here. Place specific comments wherever, preceded by "COMMENT:". Only a person designated by the Syntax Committee may insert or change statements not preceded by "COMMENT:". We can read your statement signature, and will consider how we should change our proposal in response, and then will mark "CONSIDERED" in front of your comment, possibly adding more comments.

CONVENTIONS used in this file:

A word that begins with an uppercase letter and is followed by lowercase letters is a command word. Noise words are in parentheses. Parsefunctions and rules are in all caps. Comments are interspersed between percent signs. Ignore structure in order to follow the sequence of steps in a command. Where a choice can be made by the user, there will be either (1) a comment "%one of the following:%" and the plex at the next NLS statement (ignoring more comments) is a list of the alternatives, or (2) the alternatives separated by slashes.

Commands for specifying mail to be sent:

Send Mail % keep going... %

(author:) LSEL

% include what's in the upstatement only if Author field is null %

CHANGE CONSIDERED: More people (so far) seem to like Author rather than From, so we have changed it. Note that there is another field named Sender.

COMMENT: I don't understand. I will be VERY unhappy if I have to specify the author on everything I send. If I don't say anything by the time i get here it should assume I am the author. If I said something to the contrary, it doesn't need to ask me here does it? This is augmentation?

(to:) LSEL (cc:) LSEL

% include only if both To & Cc fields are null %

(subject:) LSEL

% include only if Subject field is null %

(body of mail:) % include only if Body field is null %

% one of the following: %

Text LSEL

Statement (at) SSEL

Branch (at) SSEL

Group (at) SSEL

Plex (at) SSEL

File DSEL % BUG anywhere in the file desired, or type file address %

Rest (of file)

(to be recorded in Journal?)

% include only if user hasn't specified something for the contents of the Recorded field. This syntax assumes that there is some way for the user to specify that an item is to be unrecorded other than leaving the Recorded field null. We suggest that the Recorded field be able to contain NO (for unrecorded) or the journaldesignator (for recorded) %

% one of the following: %

OK % fill field with default journaldesignator %

Yes % fill field with default journaldesignator %

No % fill field with NO %

OPT (journal designator) LSEL

(show mail to be sent?) OK/Yes/No % If Yes or OK: display filled fields %

% only if any of the above Author, To, Cc, Subject, Body, Recorded queries were necessary %

COMMENT: Whenever possible the mail item should be on the screen for display users, and this question should not be asked then, obviously.

(send the mail?)

% only if any of the above Author, To, Cc, Subject, Body, Recorded queries were necessary %

% one of the following: %

No (the mail is still in progress)

% Send command is ended at this point. The mail remains "in progress". %

Yes

OK % If Yes or OK, keep going... %

OK / Immediate (delivery) / Deferred (delivery) OK

% The EOP R?P rquires that Immediate be the default for just OK. This should be changeable; we wouldn't want to lead most users to have to sit there and wait for the mail to actually be delivered before they can go on to the next command. %

% Note that this may immediately follow Send Mail if none of the above (Author, To, Cc, Subject, Body, Recorded) queries were necessary. If any of them were necessary, then this follows a Yes or OK answer to the previous query "(send the mail?)" %

% FUNCTIONS OF THE SEND COMMAND (above)

Send may function as an interrogation for the user who wishes to start specifying a brand-new piece of mail, see how the system understands what s/he's specified, and then dispatch the mail. In this case s/he will be prompted for all the above (Author, To, ... Send the mail?).

Or, it may be used to simply dispatch a piece of mail that the user has fully specified by using Fill commands or a mail form. Fully specified means that none of the fields mentioned above (Author through Recorded) were empty (except that if To is filled, Cc can be empty, and vice versa). In this case, the command will look like this:

Send Mail Immediate (delivery) OK or

Send Mail)eferred (delivery) JK or

Send Mail OK

Thirdly, it may be used after the user has specified some of the needed fields via the Fill command or a mail form, to check that all the required fields are filled, give him a chance to fill them or skip them, check the current state, and then ask him if he wishes the mail sent now. If he does not wish it sent, he has simply filled more fields, and the mail remains "in progress". %

COMMENT: (RLL msg of 16 May 1711) ... Lastly, I think the SEND MAIL command is great; really good job. Will there be a way to point to some 'old' mail to be sent rather than the implied current one? What about the question of pointing to a user file with the mail for in it? ...

COMMENT: The Send command works on the current mail in progress only. We do imagine some command(s) to bring another unsent item in (from a user file or a system file) to be the current one, and setting the old one aside, where you can pick it up later. However, it doesn't look like we have to have this feature ready for the benchmarks right now, so we can specify it later.

Fill (in)

% one of the following settable fields: %

Author LSEL

CHANGE CONSIDERED: More people (so far) seem to like Author rather than From, so we have changed it. Note that there is another field named Sender.

Subject LSEL

Reply (to) LSEL

Acknowledge (receipt) No/Yes/OK

% ... unfinished; we especially need to agree on what the syntax will be for filling the Recorded field %

% The Fill command replaces the current specification of the named field of the piece of mail in progress %

Append (to)

% one of the following appendable fields: %

Author LSEL

CHANGE CONSIDERED: More people (so far) seem to like Author rather than From, so we have changed it. Note that there is another field named Sender.

Subject LSEL

Reply (to) LSEL

% ... unfinished %

% The Append command adds something to the current specification (at the end) of the named field of the mail in progress %

Show

COMMENT: Isn't the AUGMENT convention to require a final, confirming OK before executing the command? Shouldn't there be a final, confirming OK before showing anything (e.g., Show Mail (in progress) OK)?

% one of the following: %

Mail (in progress)

% displays status of all fields not null for the mail in progress. Note that fields RECORDED, PRIVATE, and ACKNOWLEDGE must accept a NO specification from the user, which is different from the null in making decisions on whether the user has specified the field. The default for a null field will be NO in most cases, but we want to distinguish that he said NO. %

Author

% displays status of the Author field for the mail in progress. %

CHANGE CONSIDERED: More people (so far) seem to like Author rather than From, so we have changed it. Note that there is another field named Sender.

Subject

Reply (to)

% ... unfinished list of fields %

COMMENT: (RLL msg of 16 May 1711) ... I am very opposed to the SHOW command since it varies all over the place in whether it is recognized as S, <>S, <>SH. How about Display? View? ...

COMMENT: Suggest a better word? I assume you are proposing changing it in BASE as well. Display is inappropriate because the word has to make sense in Typewriter mode too. It does seem to me that most subsystems will have at least 3 commands beginning with S (We've already got three in our current talks about MAIL so far), so Show is usually <>SH. Is the reason for your serious important enough to mean that we shouldn't have any particular word that starts with S (or C or R) occur in all subsystems if it's not always top level?

Initialize Mail (in progress) OK

% Set the status of all fields of the "mail in progress" back to defaults (most empty). %

COMMENT: (RLL msg of 16 May 1711) ... Second, are we changing RESET to Initialize? If so we need to change the BASE command as well. Since there are many 'R' commands that might be a good idea. ...

COMMENT: No, I don't think of it as changing RESET. This command has a different function than the Set/Reset commands in other subsystems. I imagine there will ALSO be Set/Reset in this Mail subsystem, used for Mail useroptions, ie, switches for certain background settings. This Initialize command

actually erases your current work. It is central to the actually function you are performing in this subsystem. On the other hand, Reset should be used for changing the settings of certain features for the user, things that have to do with how the functional commands work, and should not be used for a command that actually does the work. For example, in BASE, you wouldn't use the command word Reset to delete modifications (comparable to Initialize in MAIL). You do use Reset to change the Viewspecs or the Name delimiters, and then use other commands to actually display or print according to those viewspecs, or to actually insert a statement name.

Commands for manipulating mail that has been received:

% CONCEPT: Status of received mail

Each item of mail that a user receives has a status. This may be implemented by storing all mail items with the same status in a plex, e.g. there would be a new branch, old branch, reminder branch, etc.

COMMENT: This presents one way of handling mail. I have another (e.g. I don't want an author branch and i would not use an old mail branch). I am sure others have different schemes. I suggest that the thing be IMPLEMENTED so that only new mail is a category that the system requires. The DEFAULTs and TRAINING can be slanted to this scheme if desired, where old mail, reminder and author are pre- defined categories. The reminder idea is great and can be user-dependent as well (e.g. "set category <name> to be reminder mail"). I think if the user uses READ to read new mail and does not move it anywhere it should be moved to a category-- again user defined and identified as a place to put read mail, but not a built in name. Such an implementatio scheme make the system more general and flexible but can look exactly like you want too.

New mail (unprocessed)

This means that the mail item has been copied into the user's initial file, but the user has never read, printed, forwarded, answered, stored, or otherwise "processed" this mail item in the MAIL subystem. An item can be processed by the Process command, as well as the Forward, Read, etc., commands.

Old mail (processed but not categorized)

The mail item has been processed in some way (ie, is no longer "new" mail), and not marked as a "reminder" item or filed under some specific category. It can be thought of as being in a miscellaneous storage area.

Reminder items ("remind me to do something with this")

The user has specified that s/he wishes the item placed in this category so s/he can ask for these items as a specific

class of items that still need to be processed, or treat both this category and the "new" mail together as "pending" mail. Some users will think of these items as "high priority" or just things s/he hasn't yet decided what to do with.

Author

A copy of every mail item you author will automatically be delivered here; you are NOT required to specify that you wish an author copy. Senders do NOT automatically receive a copy.

user's category

The mail item has been filed in a special category set up by the user, e.g. mail about the Mail subystem, mail about TOPS20.

The user may also file a mail item anywhere else he chooses just by indicating the location. %

% common rules %

MAILTYPE =

% one of the following: %

New (mail) [<OPT> (selected by) MODIFIER] % MODIFIER defined below %

Reminder (mail) [<OPT> (selected by) MODIFIER]

COMMENT: To clearify my suggestion: this be a pseudoname for the user defined category that is set to be the reminder category, rather than being a built-in category.

Pending (new and reminder mail) [<OPT> (selected by) MODIFIER]

% combination of above two categories %

Old (mail) [<OPT> (selected by) MODIFIER]

Author (copies of mail) [<OPT> (selected by) MODIFIER]

COMMENTS: Suggest Old and Author be omitted as command words here (and appear as usercategory if defined).

USERCATEGORY [<OPT> (selected by) MODIFIER]

Branch (of mail at) DSEL [<OPT> (selected by) MODIFIER]

% One has to address some branch with this construction if it hasn't been named as a special category in MAIL. %

COMMENT: I don't see any restrictions associated with removing this. And lots of gains. Mail should be kept in places that are categories, no?

Item LSEL

% "Item" refers to a single piece of mail whose citation has been delivered into the user's initial file. Only the user's mail file(s) that are known to the MAIL Subsystem are searched for this item. The catalogs of recorded mail are NOT searched (see below for Recorded mail).

I'm using citation to refer to the user-definable parts of an item that are delivered to the user's initial file. The default probably includes some parts of the item's header, plus a reference to a recorded item, or the complete text of an unrecorded item.

For LSEL, the user types or points to one or more message identifiers, separated by spaces or commas (just like an IDENTLIST).

How about some appropriate noise words after "Item" above to indicate to the user that message IDs are expected. Suggestions? %

MODIFIER =

% one of the following: (When a MODIFIER is followed by another MODIFIER, then the logical connector "AND" is assumed) %

Date (sent) Since LSEL MODIFIER/OK

Date (sent) Before LSEL MODIFIER/OK

% If Since is not also specified, Since the beginning of time is assumed %

Date (sent) Between LSEL (and) LSEL MODIFIER/OK

To PATTERN-SINGLEFIELD MODIFIER/CONFIRM

CONFIRM = % one of the following: %

OK % when some Date field is described %

OK (searching since DATE only) OK % when no Date field is described (DATE in the noise words is the date six months ago) %

Cc PATTERN-SINGLEFIELD MODIFIER/CONFIRM

Bcc PATTERN-SINGLEFIELD MODIFIER/CONFIRM

Author PATTERN-SINGLEFIELD MODIFIER/CONFIRM

Subject (word) PATTERN-SINGLEFIELD MODIFIER/CONFIRM

Keyword PATTERN-SINGLEFIELD MODIFIER/CONFIRM

Subcollection PATTERN-SINGLEFIELD MODIFIER/CONFIRM

Number LSEL OK

List of message IDs

COMMENT: We must decide whether we are going to call it a message "number" or "identifier" and then use that terminology consistently (in command words, noise words, and documentation). If it isn't entirely digits, I prefer calling it an "identifier".

Pattern PATTERN-WITHFIELDS OK

PATTERN-SINGLEFIELD = LSEL

% one or more idents, words, or whatever the command expects, separated by / for "OR", & for "AND", and grouped by parentheses %

PATTERN-WITHFIELDS = LSEL

% Field names followed by = for "EQUALS" or # for "NOT EQUAL TO" and the value; separated by / for "OR", & for "AND", - for "NOT"; and grouped by parentheses %

COMMENT: Needs clarification -- what exactly can be separated by / and & and preceded by -? what can you do with - that you can't do with NOT EQUAL (what is the reason for also allowing -)?

PROCESSTYPE = (process)

% then one or more of the following until OK is selected: %

OK

Remind (me later) OK PROCESSTYPE

Read (entire item) VIEWSPECS % Keep going: %

% show complete text

<CTRL-O> will stop the show

In Display mode show one screenful and then prompt (Type <OK> to see more, and for last page prompt (Last page, type <OK> to continue) %

COMMENT: I do not think we should refer to "pages" of

displayed information. I think the difference between "to see more" and "to continue" is awfully subtle and may well be missed. I think this should work the way such things work in all similar places in AUGMENT; i.e., either this should say "Type <OK> to continue" and, finally, "Type <OK>", or AUGMENT should be changed however we propose here. I like the way it works currently in AUGMENT.

PROCESSTYPE

Print (entire item)

% this command should continue as in BASE Print, as if the object has been specified. Should we provide the Output capability also? %

PROCESSTYPE

Acknowledge (receipt) % one of the following %

OK

Immediate (delivery) OK

Deferred (delivery) OK

PROCESSTYPE

Answer (item)

% show brief view %

(to author and) % one of the following: %

To (list)

Cc (and to list)

(include yourself?) Y/N/OK

COMMENT: The meaning of this is not clear. Does it mean "Do you want an Author copy?" I don't seee why this should work any differently from when you send a piece of mail.

(body of answer:) % same choices as for Send %

(to be recorded in Journal?) % same choices as for Send %

(show the answer to be sent?) % same choices as for Send %

(send the answer?) % one of the following: %

Y/OK OK/Immediate/Deferred

No (fill in) % one of the following: %

COMMENT: The above reflects roughly what we want to achieve here functionally, but does not specify an appropriate syntax...needs work.

SETTABLE-FIELDS

Append APPENDABLE-FIELDS

Show

%Finally:% Send (the answer) OK/ImmediateDeferred

PROCESSTYPE

Forward (item)

% show brief view %

(to:) LSEL

(comments:) LSEL

(send the mail?) % one of the following %

OK

Immediate (delivery) OK

Deferred (delivery) OK

COMMENT: The responses to this question here should be exactly the same as the responses to this question in the Send command.

PROCESSTYPE

Store (item in) % one of the following: %

USERCATEGORY

Category (to be named) LSEL

% makes a NEW user category. Note that the user does not indicate the level. It will be inserted down a level. %

Branch (at) DSEL

% will not be made a user category for the purposes of the MAIL subsystem. Automatically inserted down a level? (If No, then add a LEVELDJUST) %

PROCESSTYPE

Delete (item) OK % then prompt for the final OK <.h> to end processing for this item %

RECORDEDMAIL =

Recorded (mail) (selected by) MODIFIER

% this command word refers to mail in the journal %

USERCATEGORY =

% one of the command words made from the mail category names that the user has already specified in a previous Store command, or by using a MAIL useroption command. %

Scan [<OPT> (and insert the brief view at) DSEL LEVELADJUST]

MAILTYPE / RECORDEDMAIL

COMMENT: Shouldn't we require a final, confirming OK before executing this command?

% then show a "Brief View" of every mail item included. In Display mode, display one screenful, then prompt (Type <OK> to see more, and for last page prompt (Last page, type <OK>). %

COMMENT: In the case of new mail, what is "first"? The most recent?

COMMENT: Current convention in all mail systems i know about is to show oldest first.

% FUNCTION: This command will present a brief view (probably a line or two of the header) of all the mail items specified,, but will not alter their status as they exist in the user's file(s). This is used only for "peeking" at the mail, not for processing it. %

Process MAILTYPE OK % Keep going: %

% show a Brief View of first item %

PROCESSTYPE

% one of the following: %

(next item) % show a Brief View % PROCESSTYPE

(last item) % show a brief view % PROCESSTYPE

% FUNCTION: to present to the user each of the items in the set,

one by one, so s/he can select one or more different processes to examine and/or dispose of EACH item. The user is "reading" each piece of mail, although we do not require the user to actually read or print it, and deciding what to do with it. Basically, for "new" mail, it will be marked or filed as "old" examined mail, or filed in some other category, or deleted. Selecting OK as the only "process" is just allowing the user to indicate "OK, I got that one, that's it".

If the user asks for New or Pending mail as MAILTYPE, any new mail that hasn't been picked up from the message.txt file will be picked up before viewing.

COMMENT: Mail that is not moved should be automatically moved to a category that is user-defined and identified to be the default category to move such mail to. This might be called "old mail" as setup for new users.

Disposing of the mail item changes its status, its category, in most cases. %

% unfinished ... (Chart of status changes for different processes) %

Remind (me later of) Item LSEL OK

% Marks or files item(s) as "reminder mail" %

Read MAILTYPE / RECORDEDMAIL VIEWSPECS % Keep going: %

% if citation does not include the text of the item, show text as well as the citation of the item.

<CTRL-O> will stop the show.

In Display mode show one screenful and then prompt (Type <OK> to see more, and for last page prompt (Last page, type <OK> to continue) %

Print MAILTYPE / RECORDEDMAIL VIEWSPECS

% this command should continue as in BASE Print, as if the object has been specified, and then print the citation and text of the item. Should we provide the Output capability also? %

Acknowledge (receipt of item) LSEL % one of the following %

OK

Immediate (delivery) OK

Deferred (delivery) OK

Answer

% one of the followoing %

Item (number) LSEL

COMMENT: The noise here should be no different than in all other occurrences of the command word "Item" in this subsystem. I am in favor of leaving out the noise. If we include noise, we should be sure it reflects the terminology we want to use (number vs. identifier). This comment also applies to "Recorded" below; I am in favor of leaving "number" out of the noise.

Recorded (mail item number) LSEL

% show brief view %

(to author and)

% one of the following %

To (list)

Cc (and to list)

(include yourself?) Y/N/JK

COMMENT: I very much want to give Answer a category here -- it would store my copy there and not mail it to me. I also think it should be hard to not include yourself. Like prompt for a category, accept a ^N default, and allow OPT to say don't send to myself.

(body of answer:) % same choices as for Send %

(to be recorded in Journal?) % same choices as for Send %

COMMENT: Would be nice to have this hidden-- default could be to record only if answering a recorded message. Could specify recording or not under the send (below).

(show the mail to be sent?) % same choices as for Send %

COMMENT: Display viewers should be looking at the thing as it is composed! This question should be unnecessary.

(send the mail?) % one of the following: %

Y/OK OK/Immediate/Deferred

No (fill in) % one of the following: %

SETTABLE-FIELDS %same as in Fill in command%

Append APPENDABLE-FIELDS %same as in Append command%

Show

%Finally:% Send (the answer) OK/ImmediateDeferred

Forward MAILTYPE / RECORDEDMAIL OK

COMMENT: The OK above seems inappropriate; compare to other commands, such as Send, Answer

% show brief view %

(to:) LSEL

(comments:) LSEL

(send the mail?) OK/Deferred/Immediate

COMMENT: The responses to this question here should be exactly the same as the responses to this question in the Send command.

Store MAILTYPE / RECORDEDMAIL OK (in) % one of the following: %

COMMENT: The OK above seems inappropriate

USERCATEGORY

% USERCATEGORY = one of the command words made from the mail Category names that the user has already specified in a previous Store command (or subcommand of Process), or by using a MAIL useroption command. %

Category (to be named) LSEL

% makes a NEW user category %

Branch (at) DSEL

% will not be made a user category for the purposes of the MIAL subsystem %

Delete MAILTYPE OK OK

COMMENT: I think there should be some noise like (really?) before the last OK as in other Delete commands

Pick (up new mail) OK

% The command word has not been chosen yet. Some possibilities other than Pick up are Get, Deliver. %

COMMENT I prefer "Get"; "Pick (up)" seems too colloquial and does not convey the proper meaning when seen in a list of command words in response to question mark.

% FUNCTION: The user may specifically choose when the new mail is to be moved to the NLS initial file from the message.txt file (to give them the choice of leaving it there and yet be able to use MAIL subsystem for any mail previously moved to NLS), but nothing

else is done to it--its status remains "new". S/he may prefer to read it in BASE, or save it for later examination with MAIL commands. %

% Should we provide for selecting only some of the mail to be picked up? %

COMMENT: It is possible to pick up (deleted, examined, unexamined) mail or a combination of these and mark what is picked up as (deleted, examined) and delete or keep the message.txt file. Be nice to get by with only one or two of the possibilities.

Print MAILTYPE VIEWSPECS % Keep going: %

The "Print" command prints mail. It has the same set of objects as the "Read" command.

AUGMENT/NLS MAIL

JAKE, 26-Jun-79 10:17 < FEINLER, NLS-MSG-CATALOG.NLS.1, > 1

< MAILDEV, CATALOG.NLS;12, >, 25-May-79 14:41 DIA ;;;;

Introduction

This is the design document for MAIL that deals with the catalog-related issues.

Requirements

There will be a catalog on each host that supports a given journal. Hosts that support multiple journals will have a catalog for each journal.

The catalog will contain information for all recorded mail for a given journal submitted from any host that supports that journal. It is highly desirable to be able to (either actually or apparently) include "old" journal items in the new catalogs for the "ARC" journal (for the purpose of searches and general references).

The catalog entries will be referenced/obtained via journal numbers (not directories and file names). It will not be necessary to keep a journal file in the same directory on different hosts, nor restore an archived file into the same directory it was originally in. [Implies that the catalog DOES NOT contain the directory information].

Catalog entries for private items will never be shown to unauthorized users.

This will be done either by making the catalog unreadable by anothing but the catalog search process, or by statement level protection.

It will be possible to expand the access list of a private catalog entry. The catalog will show who did it and how the list was expanded.

It may be routine to keep all catalogs online. However, all software must function as well as possible when there are missing catalog files.

The catalog entries will contain all information necessary for determining hit/fail on catalog searches. (Might as well just include all header fields.)

The catalogs will be maintained automatically. Catalog entries will be made by a background process that cooperates with other processes on other hosts to do the following:

Make new entries in all relevant catalogs.

Make forward pointers and/or changes when "obsoletes", "addenda", and "In-Reply-To" items are received.

Expand the access lists when appropriate.

Catalog entries should be in place when users receive citations.

General Discussion

Catalog Storage Mechamism

Some thought has been given to using a data base management system to maintain the catalogs and make searches. The current decision is to use Augment files for catalog storage. Here follows some discussion:

Coments begin with "If DBMS:" to mean that this point applies if SOMEKIND of DBMS were used for the catalogs. "IF Augment" preceeds a point that applies to using Augment to store catalogs.

If DBMS: There would have to be two kind of searches-- one thru user's local mail copies and one thru catalogs. These can be exactly the same implementation if Augment files.

If DBMS: We would have to make a programatic interface to the DBMS system, perhaps over a network, which would slow catalog references over the case where the user's program (Augment) is referencing the catalog directly. Also, the programatic interface may be difficult. In the case of Magnum it is a "batch" operation since Magnum is not interactive.

If Augment: Searches over small amounts of the catalog would be about as fast as a content search over a file. It is not clear that using a DBMS would be MUCH faster than Augment, even for large searches.

If DBMS: The Journal is a very essential part of Augment. Whatever DBMS we used, we would be married to it, and those catalogs would have to be available indefinitely. If we moved Augment to another environment, the DBMS would have to be available somehow or we would have to replace it and have several different forms of catalogs and catalog interfaces. If Augment: it is as long-lived and transportable as Augment itself.

Mail Delivery To Journal

In general there are two ways: Via regular message appended to Journal-X's message.txt file, and whole files placed somewhere for a background process to enter in Journal-X.

files

The MAIL "send" function (for recorded mail) will create journal item files for the background catalog entry process. These will be placed in a certain directory (e.g. JENTRY-X for Journal-X) with a file name like nnnn.NEW to indicate that they are new journal items to be cataloged.

In order to preserve properties, evidently all mail will be entered in the journal in this way.

Note that short journal messages should be incorperated into journal message files by the entry process, but that the cross-host distribution described below will take place before the inclusion into message files (i.e. while the short journal item is still a separate file).

The same directory will be used to distribute journal items across hosts. If a background finds a .NEW item it will (1) if it is the master host, FTP the file to all supporting hosts into the entry directory with the name nnnn.COPY; (2) if it is not the master host, FTP the file to the master host into the entry directory with the name nnnn.HOSTNAME (where HOSTNAME is the name of the host from which the file is being sent).

If the background process finds a .COPY file in the entry directory (it must be a non-master host) it will enter it in the same fashion as a .NEW file (but will not send it to the master host). If it finds a .HOSTNAME file (it must be a master host) it will FTP the file to all supporting hosts except host HOSTNAME and give them the name nnnn.COPY.

The entry directory will not be used for any other purpose.

message.txt

It would be possible to enter messages sent to a Journal-X mailbox into the journal. This would allow non-MAIL users to journalize documents easily. It could be required that users first obtain a journal number and include that as the message-id, or the journal entry process could obtain the number.

Catalog Entry

immediate and deferred: who makes the entry?

On a given host, one background process will make all entries in a given Journal's catalog. It is possible that it may make entries in all journals on that host. Mail should be delivered (by MAIL subsystem) to the journal immediately, even if the delivery is "deferred".

There should be a mechanism for MAIL processes to "poke" the catalog entry process for immediate entry. The MAIL send function could append a character to a file--the background process to look frequenty for new writes on the file. When it finds them it would enter all files in the entry directory.

entry should be there before references to it appear in users mail

ACTUALLY, can probably do catalog entry in parallel with delivery to users and rarely would a user see a citation when it was not yet cataloged. By doing immediate delivery to Journal always, plus immediate catalog when the journal sees it, we can probably do well.

how to handle short recorded messages

Short journal messages start out as a single file. The background process inserts these short items into a file containing other such messages, at catalog entry time.

how to name file containing messages.

use all but the last 2 digits of the journal #?

e.g. J234 for 234xx messages.

Could probably get by dividing by 512 instead of 100 without overflowing one NLS file.

how to determine when its full

get 100 entries max per file

how to make up name of next file

add one.

how to update MAP entries when retrieve file of messages

use file name and scan MAP for entries of type "message", fix them up.

See below about MAP.

Entry steps: (for each item)

find the file in the entry directory. handle by type:

.NEW-- FTP to master or all other hosts.

.COPY-- simple entry.

.HOSTNAME-- FTP to all other supporting hosts.

determine if should be journal file or go into file of small messages

J file

Select a directory!! **** how to know if one is full?

Rename the file and set protection if any.

Select a catalog file. (how?)

Insert entry in catalog file (and set protection if any)

Fix entry in MAP.

Find entry in MAP.

Set directory, catalog-directory, catalog-name fields.

Show it as a file, now in catalog (not reserved).

J message

Enter branch in appropriate file.

Delete original file.

Select catalog file.

Insert entry in catalog file (and set protection if any)

Fix entry in MAP.

Find entry in MAP.

Set directory, catalog-directory, catalog-name fields.

Show it as a message, now in catalog (not reserved).

Catalog contents

The catalog entry consists of one statement. The name of the statement is the journal number. Attached is a property list containing the header fields (format same as for received mail).

Statement level protection may not be necessary if all header information is not visible text (only number visible text)?

References thru catalog: the MAP

all references go "thru catalog", i.e. no direct file links

references actually use a "MAP" file (binary file)

Each word of the map file represents a recorded item. It contains integer fields for the journal directory name, catalog directory and file names, and a boolean that means must go thru catalog for some reason like access or obsoleted or added-to, and a boolean that means it is a short document within a file.

Will be several such files to avoid "long" files. Simply divide the journal number (by 256K) and use the result to get the file, remainder to get word in file.

MAP fields: (minimum number of bits shown as [n])

directory[5]: use <JOURNALdirectory> directory for file

catdir[5]: use <JCATcatdir> directory for catalog

cat[6]: use file JCATcat to find catalog entry

(journal number is statement name)

qcheck[1]: boolean, check catalog for access, indirection if TRUE

message[1]: boolean, TRUE if not a file but a message.

file name is constructed from journal number.

e.g. JMESSn where n is journal number/100.

Catalog updates

Journal items are read-only. The catalog is read-write, but is only changed by the badkground catalog entry process.

Catalog entries are changed in response to mail received. The background process recognizes:

Mail with an "In-Reply-To" field.

The item being replied to (if it is a recorded document) has its "Forward" field inserted/appended to contain the new document's number.

Mail with an "Obsoletes" field.

Each item being obsoleted gets a new field "Obsoleted" to indicate that it has been obsoleted and the new number is shown. What to do it a document is obsoleted twice? This is improper (user error)-- the newer document should be obsoleted!

Mail with an "Addenda-To" field.

Each item referenced in the Addenda field gets an "Addenda-See" field inserted/appended to show that the new document is an addenda.

Mail with an "Access-Change" field.

A single item is referenced. The access list field in that item is changed to an "Old-Access-List" field. The "Access-Change" is appended to the original list and inserted as the "Access-List" for the item. A "Forward" field is inserted/appended to reference the access change document. The access change document remains in the journal to record the details of the change.

Multi-host considerations

Catalog entries on other hosts

New entries are sent to all hosts that have catalogs for a given Journal. That keeps the catalogs up to date. Each host

can decide whether to keep the new journal item itself based on available file space, etc.

File cache

No hosts are required to keep all journal entries online. The master host is required to archive all journal entries that are not kept online. Other hosts may delete journal files at will. Retrievals will be made by moving files from the master host, after getting them from tape if necessary. Hosts will maintain their own "file cache" of recently referenced or entered journal items as they see fit, without regard to what is kept online on other hosts.

Catalogs should be kept online. However, if they are not, users will be inconvenienced by the following and no more:

Journal items that require catalog-checking, namely, obsoleted, addenda-to, private, will be unavailable.

Searches over files in that catalog file will ignore the items for which there is no catalog information (the user should be notified of this at the time of the search).

Note that the MAP files MUST remain online on all hosts to make journal item available.

File deletion

When a Journal item file is deleted, the MAP is changed to show that the directory is unknown. Other entries in the MAP are left unchanged.

When a catalog file is deleted, the MAP could be marked to show that the catalog is unknown.

Auto retrieval

A non-master host can programatically FTP a file from the master host (if its online there) in response to a request for it. Not sure of best way to interface this to the user program that wants the file.

When the file is brought online, whether from another host or from tape, a (journal-X) directory is selected. It doesn't matter much which directory. The MAP file is changed to reflect the location of the entry when the file is completely online.

If the file is one full of short messages rather than a single item, the MAP has to be fixed up for each item in the file. These will all be in one small region of the MAP file.

If a catalog file is restored, the MAP must be updated for each item in the catalog file.

Number systems

Numbers for recorded documents are obtained locally at "send" time by the MAIL subsystem. The form of a journal number is <digit-string> '- <journal-designator>.

It may be possible to include only the digit string in citations for journal X where X is the default Journal for the user. For a totally isolated Journal such as the EOP would be, this is probably highly desirable. Internally however, the journal designator will always be present.

Examples: 1234-ARC or 47651-NIC or 777-EOP.

Source of Numbers

Numbers are controlled by a number file on each host. Each word in the number file corresponds to a journal item. Numbers are assigned to journal-supporting hosts a "page" (512) at a time from master host. This is done programatically by a process on a remote host logging into the master host and running a program that returns a page number. The process "watches" the number file and keeps at least one page ahead of use. The process on the master host keeps track of which pages have been given to which host, for diagnostic purposes. The number files on remote hosts will have "holes" in them corresponding to pages that have been assigned to other hosts.

Word zero of the number file corresponds to some journal number N. Word x corresponds to journal number N+x. If the word contains zero,number is unused. If used, it contains the ident of the sender, plus a bit that indicates whether that number is reserved or really used (entry cataloged).

User settable word for the file could contain:

1. the page number for the current page with free entries.

2. the "offset" (N) to be added to numbers to get the real journal number.

A utility program could be run when the file got very large in order to keep it from becoming a "long" file. It would discard the first N pages (that had no reserved, unused numbers in them) and move all other pages down and change the offset in the user settable word.

This scheme makes it fast and easy to obtain a number and easy to find out what happened to a number, or obtain a list of reserved unused numbers and responsible individuals. Numbers will be obtained by the MAIL subsystem when it sends an item, or in response to the Reserve command.

The number file will be maintained (new pages obtained, first few pages discarded when it gets too big) by the journal catalog entry background process.

Catalog File Names

The concern here is how are new catalog files named and how are catalog searches to find out which files to search in. An overriding assumption is that catalog searches are bounded first of all by date pairs. These may be provided by the user or by the system as some kind of default.

Catalog files for Journal-X could be kept in directories <catalog-X-n> (expecting that one directory may not be enough). Catalog file names should reflect the date of the earliest entry in that file, to facilitate search bounds.

For example, search bounds could be always taken as month-year. All mail originating in a given month would go in files cat-m-y.n where m is the month, y is the year, and n is the number of catalog files required for month m (hopefully one).

Another scheme that allows the system to create a new catalog file at will (rather than at start of month) might be better. For example, catalog file names could be of the form year-month-day (all numeric) to show the date of the earliest entry in it. There may be an algorithm involving automatic file name completion that would allow fairly efficient searching for the first catalog file to search in for a retrieval.

Another scheme would be to create catalog files at will and give them sequential names like catN. A separate file would act as a key into the catalog files. In this file, month-dates would be statement names and there you would find links to the appropriate place in the appropriate catalog file to start (or stop) searching.

Processes

Data Structures

Procedures

Not for distribution

Augment/NLS Mail

JAKE, 26-Jun-79 10:31 < FEINLER, NLS-MSG-OVERALL-DESIGN.NLS.1, > 1

< ANDREWS, NEW-OVERALL.NLS;1, >, 29-May-79 14:40 DIA ;;;;

Introduction

This document describes the integrated mail system to replace SENDMAIL and MESSAGE for AUGMENT. There will be a single subsystem called MAIL that will be used for sending, reading, and manipulating both Journal and sequential mail.

It will be possible to send mail to individuals in the ident system, individuals that use network-wide sequential mail systems, or both, with the same commands. It will be possible to read mail from either type of source with the same commands.

What about other way around

This will be a multi-host, multi-journal system:

Eventually users from any number of hosts (that are linked by networks) may use the same Journal (true multi-host journal). There may be several Journals. These may be viewed as sub-journals and may be private or not connected to the primary "Journal" for various reasons, but citation numbers will be unique over all journals. The citation numbers may be different than the current Journal, but it will be possible to retrieve "old" Journal items.

There may be multiple ident systems. It would be possible to send mail to an ident in any known ident system. Hosts, Journals, and Ident systems would be independent, but each user would belong in a specific Ident system, have a default Journal, and a single host at which online mail was delivered.

User Interface Design

Mail Document Elements

The elements of a piece of mail consist of header elements plus the message (or document) body. Certain header elements are required, most are optional. The message body is not required. These fields will be present in the actual mail document, but may not be shown to the user when he sees either a brief view or a full view of the mail. The names of the fields below are as specified in the RFC733 document. The old Sendmail name is shown in parens if it existed.

The fields that are not specified in RFC733 are shown with an *. The RFC733 fields may have more specific syntax and use than specified here--these descriptions are for a general understanding--refer to RFC733 for details. The REQUIRED comment means that it is required in our context, not necessarily by RFC733.

Header Fields:

Subject (Title). REQJIRED.

Note: MAIL requires Subject fields in mail that it sends

out. It receives mail without Subject fields.

A string of free text.

From (Author). REQUIRED.

A list of addresses.

Sender (Clerk).

A single address (mailbox). Not present if same as From.

Reply-To.

A list of addresses. Replies to message to be sent here as well as (possibly) others.

RFC733 suggests that if Reply-To is present, replies not be sent to the From but only to the Reply-To addressees, automatically. The user would have a chance to add to the distribution however.

Date. REQUIRED.

Date and time message sent.

To (Distribution for action). REQUIRED.

Note: It is required that one of From, Cc, Bcc, be present.

A list of addresses.

Cc (Distribution for info).

A list of addresses.

Bcc.

A list of addresses. Only the authors and Bcc recipients will bee the Bcc field.

File-Carbon*

COMMENT: Did we agree to use my (BLP's) idea about "category" names in the To/Cc/Bcc fields instead?

COMMENT: Not fully resolved. More efficient to separate category names in a separate field like Fcc:

Desirable for EDP. What goes here and how does it work?

Message-ID (Number). REQUIRED.

RFC733 says '< (text) '>. The message-id will be formated to be a link.

In-Reply-To.

List of message identifiers to which this message replies. This field will be used to retrieve all messages relating to some recorded discussion.

References.

List of message identifiers to which this message refers.

Comment.

A text string.

Keywords.

A text string. Used for searching.

Subcollections*.

A text string. Used for searching.

Addenda-To*.

A single message-id. Used in automatic catalog maintenance and document retrieval.

Obsoletes*.

A list of message-ids. Used in automatic catalog maintenance and document retrieval.

Action-Code*.

A text string. For user purposes.

Length*. REQUIRED.

The number of characters in the message body.

Private*.

A list of idents of individuals that may access the document. If the "Private" field is not present the document is public. The access list restricts access to the recorded copy and catalog entry of the document. Unrecorded mail may have a "Private" field, but access cannot be restricted except by user control since the document will appear in full in every recipient's mail file.

Part-of*.

A single message-ID. Indicates that this document is part of (or chapter of) a "superdocument". Used in automatic catalog maintenance and document retrieval.

Original-Header*.

A message header (formatted in such a way that it permits parsing of the message header it is included in). This will be automaticaly inserted in a Forwarded item, and will be a copy of the header of that forwarded item--the message body of the forwarded item will be copied and become the body of this item.

Acknowledge-Delivery*.

If this field is present, at the time the message finally delivered to the user's mailbox, a short message of acknowledgement is sent to the Sender automatically (From if no Sender).

Date-Received*.

This field conatins the date and time of mail delivery to this user. (Field is created by mail reading process using information stored by mail delivery program). This field is not present in recorded copies.

Forward*.

A list of message-ids.

This field will ONLY be present in catalog entries, and will be inserted automatically during catalog maintenance. Means that this document is referenced by the listed documents, either as "In-Reply-To" or "Addenda-To" or "Access-Change".

Obsoleted*.

A single message-id.

This field will ONLY be present in catalog entries, and will be inserted automatically during catalog maintenance. Means that this document has been obsoleted by the mentioned document.

Addenda-See*.

A list of message-ids.

This field will ONLY be present in catalog entries, and will be inserted automatically during catalog maintenance. Means that this document has been suplemented by the mentioned document.

Access-Change*.

A list of idents. Must be accompanied by a Reference field.

This field will be used only to change the access of a private recorded document. The list of idents are added to

the access list for the document in question and the access-change document (which is also recorded) is added in the "Forward" field for the document. The old access list is saved in an "Old-Access-List" field and the new one replaces the old Private field.

Old-Access-List*.

A list of idents.

This field will appear ONLY in a catalog is will be inserted automatically. If multiple access changes have been made, the fields will be inserted in the same order that the "Forward" references are made.

Commands See (maildev,commands,)

Addresses

Addresses will be individual idents, group idents, ARPANET addresses, or Postal mail addresses. These will conform to RFC733.

Note that this means that "ARC Staff" cannot appear as a legal "To" address. It must be ARC Staff: Allen, Andrews, Beck, ...

Considerable checking has to be done on addresses of outgoing mail that is to individuals that do not use MAIL. In particular, idents have to be changed to ARPANET addresses so that other mail systems can answer the mail, host names have to be added to local addresses of mail going to another host.

COMMENT: Sug: Why not just ALWAYS send out full ArpaNet addresses so that even MAIL users can sometimes use MSG, Hermes, etc. to process their mail. Besides, it would be slightly simpler to program. Also, it could be a UserOption what to do on input from the Message.txt file, e.g.,, "contract" the addresses, convert to idents where possible.

Idents that belong to "foreign" ident systems will appear as "IDENT@HOSTNAME" where HOSTNAME is the name of the "master" host for that identsystem? This needs some work! In particular, what happens if a message is sent to a non-MAIL user and also to a foreign ident. That ident should be resolved to an ARPANET address, but that has to be done by the foreign ident system!

The current planned solution to this is to have an ident system mailbox such as IDENTSYS@HOSTNAME. A automatic deamon would forward mail to the proper mailbox when the address was in a form such as IDENTSYS (IDENT)@HOSTNAME. This address could then be used both as an address to send and a return address appearing in mail to non-MAIL users.

Mail Item Format

The format of the citation will be specified by a useroptions

Parameter. A default form will be available, or a program may be supplied by the user (?). The default form will include appropriate fields and the message body if it exists. Since the Answer, etc. comands need access to fields that the format may exclude (as well as the exact field name which may be changed), the entire header will be included in a text property that is invisible, but available to the MAIL subsystem.

COMMENT: Sug: See <MailDev,Command-Syntax,> on "Filters", Screens, and Templates".

Message-IDs

Recorded message-IDs are journal numbers. Journal numbers will include a journal-designator. The journal-designator may be hidden from the user if it is the same as his/her defaults, and then added at the time Journal numbers are referenced.

Unrecorded message-IDs will take a different form. There will be three fields: The ident, the originating host, and a sequence number.

Construct the id out of the user's ident, originating host, and a sequence number. The next available sequence number could be kept in the initials file. This would make the unrecorded numbers look at least a little like the recorded ones. And each user could keep copies of his/her unrecorded messages by sequence number alone, as he/she saw fit, for easy reference.

Thus a recorded item may have the message-id 12345-ARC and an unrecorded number may have the message-id DCE-1234C-OF3.

Forwarding

When mail item A is forwarded mail item B is generated. The body of B is a copy of the body of A. One field in the header of B contains the header of A. This permits the receivers of B to see the original header, and also see who forwarded it, who he/she sent it to, etc., as well as any comments that were added. If B is forwarded, (and becomes C) the recipients see all previous headers and comments that were added along the way.

If the item being forwarded is simply a citation for a recorded document, the message bodies are all empty and the original header contains the message-id of the document. Note that the process that formats the mail actually shown to the user probably wants to handle forwarded mail rather carefully so that the user sees the most desirable part and a minimum of info he/she does not care about.

Consider the case of a private recorded document. One of the recipients wants to forward the document and expand the access list. We have discussed two ways to handle this.

Create a new recorded item which simply has an expanded access list and points to the original (which has the original access

list). This has the rather awful side effect of creating two numbers for the same document, with some individuals being able to use only one of them.

Change the access list in the original. This violates the principle of a read-only journal in some sense, and opens the question of (1) can recipients expand the access list or only the authors, and (2) how do you protect the access list from unauthorized changes. A solution to some of this would be to keep the access list in the catalog as well as or instead of in the file. The access list in the catalog would be expandable. Somehow it must be possible to gain access through the catalog if the user is authorized, but not directly if the user is unauthroized. Thus the journal documents themselves would be read-only. By putting statement-protection on the catalog entries, unauthorizes changes of the access list could be prevented.

Catalog Lookup

All (in theory) references to recorded (Journal) items will be made through a catalog or equivalent. This will eliminate the directory name, multi-host problem. In fact the home directory for a given item will probably change from time to time on hosts that are not the "master host" for the Journal in question.

All references to updated (Addenda-To) items will automatically be informed or shown the suplement. If an item has been obsoleted, the user will be informed of that fact and shown the new document.

Overall Technical Design

Mail Delivery

Mail delivery will be the same as or parallel to the way ARPANET mail is delivered now, except that the message body will not be present for "long" recorded items.

To send deferred mail, the MAIL subsystem will create a file in the user's login directory with a name such as [--UNSENT-MAIL--]. A background mailer process will search directories for such files and, using the name extension, deliver them to the proper directory or proper host. This is currently done by MAILER. We may write our own MAILER for our own needs. We may get by using MAILER for the immediate future.

MAILER delivers mail to foreign hosts using ARPANET FTP (NIC 29588). We will write a new mailer to get mail to other hosts via Tymnet.

For immediate delivery, there are three choices. First, MAIL can perform the delivery itself. Second, it can "poke" MAILER by setting flags in a <SYSTEM> file that MAILER checks frequently for new writes. Third, a sub-fork under MAILER could do the delivery. Poking MAILER or running a sub-fork is cleaner since the code exists in only one place and the subsystem need not contain the

code. Poking has an inherent delay built into it however. It is possible to have MAIL subsystem deliver local mail but always queue foreign mail.

For purposes of determining if mail has been sent, it may be advantagous to have the message-id be part of the unsent mail file name.

The current plan

Mail will be delivered to the journal in the form of an Augment file. Thus everything will be preserved in the record. Mail delivered to users will have structure preserved. Signatures will be reconstructed to show the statement author as the mail author, time and date as the time and date the mail was sent. Users wishing to preserve properties in mail will have to record it. If the message body is short, the recipient may have to jump to the journal copy to see the properites (other than straight text).

The old plan

Mail sent to Journal for recording will always contain the full document. Mail sent to MAIL users will contain statement signatures and graphics encoded in the text in such a way that it can easily be restored by the MAIL recipient process. Mail sent to non-MAIL users will omit signatures and graphics. For this purpose, MAIL users will be those in the ident system that recieve on-line delivery. If a mail item with encoded signatures is read via MSG the user may see some slightly strange stuff. The encoding should be done so that it is as unobtrusive as possible.

Mail Receipt

Mail will be received in a users "MESSAGE.TXT" or "MAIL.TXT" file in the current format. The MAIL Read command and initialization rule, will move all MESSAGE.TXT entries into an unread mail branch and empty out the MESSAGE.TXT file.

COMMENT: Sug: It be a UserOption as to when, if at all, Messge.txt gets automatically Moved to the user's .NLS file.

There will be automatic notification of receipt of new mail (a message in status window). Apparently this has to be done by the FE. It may or may not be possible/desirable to automatically "read" the mail at that time. It is somewhat feasible to go into a special command IF the user is in MAIL at the time new mail is noticed. If the user is in another subsystem it is not clear whether this is desirable or always possible.

COMMENT: I don't think it desirable to "read" in new mail when in a subsystem other than MAIL. I think it should be a UserOption as to whether to do it automatically when in MAIL.

Current plan is to read message.txt mail only in response to

commands in MAIL subsystem.

Users will receive complete documents for unrecorded mail. Recorded mail will be handled like Journal items are now--short messages appear in full, others are just a citation (a message without body in this context).

The formatting of the incoming messages into AUGMENT text will be designed in such a way that an arbitrary program can be patched into the process. The entire message header will be included as a property, which can be formatted for easy parsing by program, as the user will never see it directly.

Journal Entry See (maildev,catalog,) for this and related issues.

Cross Host transfer

Mail going to another host will follow the ARPANET protocols. For Tymnet we will use Telecopy if adequate, or establish our own protocol.

For a given Journal, there will be a single host that is "master" host for that Journal. All Journal entries will be sent to that host. Local copies will be kept on other hosts as required (recently referenced/entered items will exist online in a "cache" of files). The other hosts will generally have complete catalogs, but there should be a mechanism for referencing items from a host that has no catalogs for the specified Journal. There should be an automatic mechanism for retrieving Journal files on the Journal host and FTPing them to a desired host. The files to be retrieved will be referenced by journal number and no further qualifications (such as directory).

Numbers and Catalog References See (maildev,catalog,)

Early Command Capabilities Descriptions

Mail Originating Commands (Capabilities, not command specifications)

Interrogate (necessary fields)

Like SENDMAIL's interrogate Asks user for required fields, then allows for additional fields to be spcified.

Specify <specify individual fields>

Allows specification of any fields named above, plus specification of the message body. Automatic repeat of "Specify" part of command, user types command word for field to specify. Like SENDMAIL's Authors, Comment, Number, etc. commands.

This may be the proper place to specify whether the item is to be recorded (with an option to specify which journal) or unrecorded.

Send (immediate/deferred)

Like SENDMAIL Send. Immediate takes longer since mail is actually sent. When Recorded mail is sent Immediate, cataloging is "done" when Send command is finished. Deferred is similar to SENDMAIL's Send or SNDMSG Queued. It may be that the recorded/unrecorded option is specified in this command.

Send WILL NOT proceed if the required fields have not been specified.

Save/Restore status

Like SENDMAIL Insert Status and Process.

Reserve number

(optionally takes a journal designator to show which journal).

Recorded (optional Journal identifier)/Unrecorded

Recorded documents are entered in the Journal. Unrecorded documents are not entered and the message-id is different; recipients will always receive the actual document, not a citation.

Show status

Like SENDMAIL's.

Mail Manipulating Commands

Model of how MAIL is used:

When entering MAIL and when the BriefView command is given, new mail is entered into a special branch in the initials file.

The user then gets a "brief view" of the new mail he/she has not seen before. Under mail-option switch control, subsequent "brief views" may or may not show those same items again. (I.e. in one case Brief view only shows unseen (by brief view) mail items, in the other it shows unread mail).

The user then reads his/her new mail (in one of two ways) and does one of the following with each item: delete or move item to another branch of mail items. Before or after a Move, the user can Answer and/or Forward. (I.e. user can Answer/Forward any mail item, not just new mail). The move command takes a category name and moves the item into that category (most recent first). Categories are branches in the initials file. These branches may contain the actual mail items, or links pointing to files that contain mail items. The Move command will allow the user to make up a new catagory at the time the command is specified (probably via the option key). The Move command will, under mail-option switch, recognize (old) categories as command words, or take them as full text

specifications. (The thought here is that the user may have so many catagories that command-words are not useful).

COMMENT: The "full text" specification is guaranteed to take AT LEAST as many keystrokes as a command-word (and usually more) and a user would find out later that he/she had mistyped. Hence I see no reason for the "full text" option.

COMMENT: The "full text" option means that the user with lots of categories does not have to remember how many chars each takes to trip the recognition, nor wait for the system to see if he/she typed enough-- just type the whole thing.. Mistyping is not the issue.

COMMENT: Ahh -- I see. I agree that there ought to be a "full text" option.

The mail in the "new mail" branch (that has not been moved or deleted) will be given statement names such as m1, m2, etc. These will be used in the manipulation commands as names. After the mail is moved, the name will become the message-id. Then the mail item must be bugged or the message-id given in order to manipulate it further.

The delete command REALLY deletes the item and requires two confirms. (The user can easily move to a deleted category that can be expunged at will if that is what is desired).

The two ways to read the mail are (1) to use the Read command which cycles through the new mail items interrogate-fashion and allows the user to dispose of each in turn until he/she uses CD or runs out of new mail, and (2) by using top level commands to Move or Delete items by addressing them (e.g. m1, m2, ...).

On a display, the brief view will remain on the screen to permit bugging while handing mail. When an item is shown, it covers the brief view, but the brief view is restored when the item is Moved or Deleted.

Read New (mail)

Enters a command loop that allows user to handle new mail with minimum of typing (user may CD out of loop and enter it again later). The user will not be prompted for WHICH message to (Move, Delete, Answer, Forward, etc). A mail-option switch will control whether or not default action is taken after the item is read. This action, if taken, will be specified as a sequence of commands, for example "Move (to) old-mail".

The mail-option switches specified so far should be enough to satisfy almost everyone's desires for catagories of seen/unseen/recent/old mail.

Skip

This command will be available in the "Read" command loop only. It moves current piece of mail to end of unread mail list for later processing.

Answer

Enables user to reply to current piece of mail, ala MSG. Automatically fills in an "In-Reply-To" field.

Forward

Forwards piece of mail with optional comments to be added. These comments appear in the Comment field of the item sent. The header of the item being forwarded is autmatically included in the "Previous header" field. See discussion below about forwarding private documents.

Delete

REALLY deletes a mail item. Requires two confirms.

Move

Moves current or selected piece of mail to any category. May create a new category.

Copy (useful?)

Copies current or selected piece of mail to any category. May create a new category.

Expand (useful?)

For a citation only, inserts the document body so the user has a copy in his own file space. (The number of characters in the document is shown in the citation, or show to the user and get confirm?).

Acknowledge receipt (useful?)

Sends short message acknowledging receipt of current message, to Sender (or From) only.

Output (new mail)

Prints (expanded form if requested) the unread mail branch.

Output (mail branch)

Prints any branch of mail items (expanded if requested).

Output (mail item)

Prints only one mail item.

Jump Link

Jump Link on a message-id of a recorded item will get user to the recorded document if it is online, as it does now. Note that this will always reference the Journal number, never a regular file link. We will make an effort to make this nearly as fast as the current jump on a directory,file journal reference.

Survey (mail branch)

This command will perform searches over given mail branches or catalogs and give a brief view of "passing" documents. On a display, the brief view of documents will be accessible via Jump Return or some such mechanism, so that the user can jump to selected items and return to the brief view, etc.

A query phrase specified in the command will include "ALL", <field> = <text> or <field> CONTAINS <text>, and boolean operators involving these tests.

START User FEINLER Job NLS Seq. 7748 Date 26-Jun-79 10:33:29 Monitor SRI-K

```
FFFFFFFFFF    EEEEEEEEEE      IIIIII      NN      NN    LL            EEEEEEEEE
FFFFFFFFFF    EEEEEEEEEE      IIIIII      NN      NN    LL            EEEEEEEEE
FF            EE                II        NN      NN    LL            EE
FF            EE                II        NN      NN    LL            EE
FF            EE                II        NNNN    NN    LL            EE
FF            EE                II        NNNN    NN    LL            EE
FFFFFFFF      EEEEEEEE          II        NN  NN  NN    LL            EEEEEEEE
FFFFFFFF      EEEEEEEE          II        NN  NN  NN    LL            EEEEEEEE
FF            EE                II        NN    NNNN    LL            EE
FF            EE                II        NN    NNNN    LL            EE
FF            EE                II        NN      NN    LL            EE
FF            EE                II        NN      NN    LL            EE
FF            EEEEEEEEEE      IIIIII      NN      NN    LLLLLLLLLL    EEEEEEEEE
FF            EEEEEEEEEE      IIIIII      NN      NN    LLLLLLLLLL    EEEEEEEEE
```

START User FEINLER Job NLS Seq. 7748 Date 26-Jun-79 10:33:29 Monitor SRI-K

START User FEINLER Job NLS Seq. 7748 Date 26-Jun-79 10:33:29 Monitor SRI-K

START User FEINLER Job NLS Seq. 7748 Date 26-Jun-79 10:33:29 Monitor SRI-K

START User FEINLER Job NLS Seq. 7748 Date 26-Jun-79 10:33:29 Monitor SRI-K

START User FEINLER Job NLS Seq. 7748 Date 26-Jun-79 10:33:29 Monitor SRI-K

```
0000000001111111111222222222233333333334444444444555555555566666666667777777777
1234567890123456789012345678901234567890123456789012345678901234567890123456789
 *START* User FEINLER  Job  NLS Seq. 7748 Date 26-Jun-79 10:33:29 Monitor SRI-K
```

```
FFFFFFFFFF    EEEEEEEEEE     IIIIII     NN      NN    LL            EEEEEEEEE
FFFFFFFFFF    EEEEEEEEEE     IIIIII     NN      NN    LL            EEEEEEEEE
FF            EE               II       NN      NN    LL            EE
FF            EE               II       NN      NN    LL            EE
FF            EE               II       NNNN    NN    LL            EE
FF            EE               II       NNNN    NN    LL            EE
FFFFFFFF      EEEEEEEE         II       NN  NN  NN    LL            EEEEEEEE
FFFFFFFF      EEEEEEEE         II       NN  NN  NN    LL            EEEEEEEE
FF            EE               II       NN    NNNN    LL            EE
FF            EE               II       NN    NNNN    LL            EE
FF            EE               II       NN      NN    LL            EE
FF            EE               II       NN      NN    LL            EE
FF            EEEEEEEEEE     IIIIII     NN      NN    LLLLLLLLLL    EEEEEEEEE
FF            EEEEEEEEEE     IIIIII     NN      NN    LLLLLLLLLL    EEEEEEEEE
```

START User FEINLER Job NLS Seq. 7748 Date 26-Jun-79 10:33:29 Monitor SRI-K

START User FEINLER Job NLS Seq. 7748 Date 26-Jun-79 10:33:29 Monitor SRI-K

START User FEINLER Job NLS Seq. 7748 Date 26-Jun-79 10:33:29 Monitor SRI-K

START User FEINLER Job NLS Seq. 7748 Date 26-Jun-79 10:33:29 Monitor SRI-K

START User FEINLER Job NLS Seq. 7748 Date 26-Jun-79 10:33:29 Monitor SRI-K

START User FEINLER Job NLS Seq. 7748 Date 26-Jun-79 10:33:29 Monitor SRI-K

< MAILDEV, MAILSUPPORT.NLS;32, >, 30-May-79 09:09 MEB3 ;;;;

Introduction

Design document for the parts of MAIL subsystem that interface to MESSAGE.TXT file, RFC733 format, and mail delivery.

GENERAL COMMENTS by BROWSER'S -

Place general comments here.

Now we're getting there! I have added extensive comments, but they to not have a great impact.

Place specific comments anywhere preceding it with "COMMENT".

Outigoing Mail

General

It is assumed that the message to be sent has been created and exists in an AUGMENT file as a branch. It should be in a form in which it is easy to get a handle on individual fields in the header (some crucial ones being to, cc, bcc, reply-to) and the body.

I strongly suggest that each field in the header be a separate statement. Sending the item will be much easier if the mailing procedure does not have to do a lot of string analyzing in order to isolate individual headers.

The address fields of a message sent to non-Mail user's will be arpanetized (i.e., all idents will be transformed into arpanet mailboxes). The address fields of a message sent to a Mail user will be exactly the same as it appears in the composed message (i.e., a mixture of idents and arpanet mailboxes). The assumption at send time will be that idents imply mail user and arpanet mailboxes imply non-Mail user. If the arpanet mailbox of a mail user is to be transformed into an ident, it should be done at message composition time.

COMMENT: The code should be organized so that the mail user/non mail user info can be extracted from the ident system at a later time (when the ident system is improved).

Step1: Add Date and Message-ID fields.

Get current date and time, and add Date field to AUGMENT file.

The Message-ID is obtained by calling a procedure (to be specifed by Don) with arguments: ident of sender, recorded boolean. Make up Message-ID field and add it to AUGMENT file.

Step2: If the message is recorded deliver message to Journal.

Call a procedure which creates an AUGMENT file in directory ?????

(ask Don) consisting of a copy of the header appearing in the mail branch without the bcc field and the full body of the message.

COMMENT: The header field property will not be included at this time (it goes in the catalog).

Step3: Get arpanet mailboxes which will receive the messages.

Four strings will be constructed:

MSENDTOCC - Arpnet mailboxes of all addressees in to and cc fields who are Mail users separated by commas.

MSENDBCC - Arpnet mailboxes of all addressees in bcc field who are Mail users separated by commas.

NMSENDTOCC - Arpnet mailboxes of all addressees in to and cc fields who are Non-Mail users separated by commas.

NMSENDBCC - Arpnet mailboxes of all addressees in bcc field who are Non-Mail users separated by commas.

It will be useful to construct three more strings at this time.

ARPATO - Arpanet mailboxes of all addressees appearing in to field.

ARPACC - Arpanet mailboxes of all addressees appearing in cc field.

ARPABCC - Arpanet mailboxes of all addressees appearing in bcc field.

The ARPA* strings will be used to replace the to, cc and bcc fields in the header for any mail sent to a Non-Mail user.

While all this is going on we can also set flags called toMail (TRUE if item going to at least one Mail user) and toNon-Mail (TRUE if item going to at least one Non-Mail user).

This is done by examining the to, cc and bcc fields in the AUGMENT file. When a mailbox is encountered it is added to the appropriate nmsend string. When an individual ident is encountered we must look up the record for that ident in the IDENT file get the user name and host where he receives his mail construct the mailbox string (<user>@<hostname>) and append it to the appropriate msend string. When a group ident is encountered we must resolve it to individual idents and build mailbox strings for those idents.

COMMENT: More detail on ident lookup: When a phrase is encountered, if it MIGHT be an ident (5 chars or less) look up in the ident system. If there OK, if not see if user name (via DIRST JSYS). If not user name, get help from the user! If longer than 5 chars, check for user name and if fail, get help from user. If the phrase is followed by a machine-id (@OFFICE-3) assume the name is a correct mailbox on that

machine. The hook for multiple idents goes in here somewhere.

There are two existing procedures in the ELITE stuff called cvdirid and cresolvid that go a long way towards accomplishing all this.

While the msendto and the msendcc fields could be combined into one string (likewise for nmsendto and nmsendcc) for the prurposes of mailing it is better to do this separately since we can use these results to replace the separate fields for messages sent to non-Mail users at the appropriate time.

LATER CONSIDERATION: If we ever have multiple ident systems we need to consider addresses of the form "<ident>in<identsystemname>".

Step4: Construct temporary files containing various parts of the message.

If there are any Mail receivers of the message construct a temporary sequential file called [--MAIL-ADDRESS-PART--]. It will contain the following fields of the header:

Date field

From field

Subject field

Sender field (if present)

Reply-to field (if present)

To field

Cc field (if present)

If there are any Non-Mail receivers of the message construct a temporary sequential file called [--NON-MAIL-ADDRESS-PART--]. It will contain the same fields as [--MAIL-ADDRESS-PART--] but the values of the address fields will contain arpanetized addresses. The values for the to, cc and bcc fields are contained in ARPATO, ARPACC and ARPABCC respectively. We must go to the IDENT file again to arpanetize the From, Sender and Reply-To fields.

Construct a temporary sequential file called [--REST-OF-HEADER--]. It will contain the remaining fields of the header.

If the message is unrecorded construct a temporary sequential file called [--MESSAGE-BODY--]. It will contain the sequentialized body of the message with encoded structural information.

If the message is recorded construct the [--MESSAGE-BODY--] file containing the sequentialized body of the message with encoded structural information if:

1) the length of the message is less than n, or

2) the message is going to a Non-Mail user.

Encode structure by changing all EOL characters to CRLF and indicating statement breaks by appending CRLF followed by n+1 CR's where n represents the relative level of the statement in the message. Two CR's represent top level statement.

Step5: Send it to all addresseees.

Unrecorded mail

The message will be delivered to each mailbox found in the strings MSENDTOCC, NMSENDTOCC, MSENDBCC and NMSENDBCC strings.

The basic algorithm for sending unrecorded messages has only slight variations according to the address string in which it is found. These variations are indicated in the algorithm at the point where they occur. For each mailbox in MSENDTOCC and NMSENDTOCC do:

a) If mailbox is local attempt to append message to appropriate message.txt file. This involves appending three files to message.txt: Either [--MAIL-ADDRESS-PART--] (if mailbox came from MSENDTOCC) or [--NON-MAIL-ADDRESS-PART] (if mailbox came from NMSENDTOCC), [--REST-OF-HEADER--] and [--MESSAGE-BODY--].

b) Create a sequential file called [--UNSENT-MAIL--].<mailbox>.

c) Append the contents of Either [--MAIL-ADDRESS-PART--] (if mailbox came from MSENDTOCC) or [--NON-MAIL-ADDRESS-PART] (if mailbox came from NMSENDTOCC), [--REST-OF-HEADER--] and [--MESSAGE-BODY--]

d) Append the sequential version of the bcc string found in the AUGMENT file to [--MAIL-ADDRESS-PART--] and append ARPABCC string to the [--NON-MAIL-ADDRESS-PART] file.

e) For each mailbox in MSENDBCC and NMSENDBCC do a) b) and c).

NOTE: I talked to RLL about the semantics of the bcc field and we agreed that anyone listed in the bcc field should receive the entire bcc address list.

I envission a single procedure to do all the appending of files. Its arguments will be:

tojfn - jfn of the file being appended to

apjfn - jfn of file containing address part of header

rhjfn - jfn of file containing the rest of the header (and

null line)

mbjfn - jfn of file containing the body or zero if the body is not to be included.

Recorded mail

This is the same as unrecorded mail with the exception that sometimes the body will be included and sometimes not.

Step6: Poke mailer to deliver [--UNSENT-MAIL--] for immediate delivery.

Incoming Mail

Reading MESSAGE.TXT File

General discussion

All messages that appear in the message.txt file at the time it is read will be examined under user option control. Each message that is undeleted will be moved (message.txt will be deleted in this case) or copied (each copied message will be marked deleted in this case).

Reading the file

The first line will be read in. This line (hereafter called the special line) contains the date and time recieved followed by a comma, number of characters in the message, followed by a semicolon, followed by twelve characters used by msg to mark the message and precedes the actual mail item. The mail item which follows will be processed as described below. We then read the next special line and proceed until the file has been exhausted.

See procedure MINMES in the Message subsystem for help here. It cannot be used as is of course.

Reading an individual message

Examine the special line extracting the number of characters in the message and how it is marked.

If the message is marked deleted (a 2 or 3 will appear in the twelfth character of the area used by msg to do marking) skip to the next special line (this is done by reading the number of characters specified in the current special line).

Once it has been decided that a message is to be copied/moved into the new mail branch insert a statement down from the new mail branch with "(Mn)" as the text for this statement. 'n will be an integer one greater than the integer appearing in the previous entry.

After the above statement has been created the entire header

will be placed into this statement as textual properties. Each header field is a line of sequence of characters followed by CRLF followed by a non-space or nonhorizontal tab character. Each header field consist of a field-name folloed by a colon followed by the fieldvalue.Each field is entered into the statement as a separate property which is identified by the field-name as follows:

The name of the property will be the field-name and the property will consist of a string header followed by block of characters specified in the field-value. The one exception will be the field containing the header of a forwarded message. This property will be an inferior tree containing the properties of the forwarded header.

The routine used to create all properties except the inferior tree property is creprop. All we need to do here is create a string, determine the number of words used in the string data structure and call creprop.

kThe routine used to create inferior trees is creit. All we need to do it is provide an stid of the statement to contain the property and the property type. It returns 0 if error or stid of origin of inferior tree.

E.G. creprop(stid, fieldtype, size, $fieldvalue)

stid - stid of statement to contain the property

fieldtype - this is a predefined type. It is declared as a global constant with a value in the range 40000B-777777B.

length - length of the string data structure

fieldvalue - name of the string containing value to put into property (the array at $fieldvalue of length size)

NOTE: This means that the names of all properties must be known ahead of time. Thus I suggest another property be specified which will contain all fields not known to the Mail subsystem. They would only be displayed if the user wanted to see all fields in the header.

To get a property one must load it by calling lodprop

E.G., lodprop(stid, proptype)returns FALSE if error, page number in core if success and address of block in core. Address plus five gets the string stored there . The block must be frozen if one is to do any kind of analysis with it .

COMMENT: Watch out for non-RFC733 messages.

The text of the statement the user will see as the header is then formatted using a default format or user designated format. The header property will be the place where the formatter will find the desired items.

The body of a message not composed in the Mail subsystem (has no encoded structure imbedded in the message) is copied underneath the header after the body has been AUGMENTized. How about just doing a copy sequential type of thing?

The body of a message composed in the Mail subsystem (has encoded structure imbedded in the message) is then handled as follows:

1. If the message is unrecorded then the body is moved into the AUGMENT file and restructured according to the encoding scheme discussed in Outgoing Mail.

2. If the message is recorded and the length is less than a user option parameter it is moved underneath the header in the AUGMENT file as in 1.

3. If the message is a "long" recorded message the body is not entered into the initials file. HELP! Is body then entered into local Journal in case of cross host mailing?

COMMENT: When Expanding a Journal item Body, there is no guarentee that the Body is on-line anywhere, much less that it is on-line at the local host. Sug: The routine the Moves/Copies/Expands Bodies first try to find the Body in the appropriate Journal on the local host; if that fails, try the appropriate "master-Journal", if no-local; failing that, find out where the item is archived and ask the user if he/she wishes a retrieval to be initiated.

NB: The above is probably a two cycle operation as the catalog entry has to be found before the Body can be found and I think you have to go thru all the rigamarole above to fine the catalog entry also.

COMMENT: the above should only be done in response to a comand, (e.g. Jump or Expand) NOT getting mail out of message.txt and into initials file!

Areas to Consider and Questions

The user may choose to break the header statement and later decide to delete one of the statements. If the statement he deletes contains the header property he will unknowling delete any chance of answering or forwarding or reformatting the message.

finding message length

COMMENT: the message length is in the message.txt file.

Each message is preceeded by a line of garbage-like text that has the time of delivery, message size, and code about seen,examined, etc.

restructuring and attaching signatures

COMMENT: restructuring the (relative) level information

giving pointers to the message to internal formatter

COMMENT: a stid should serve

COMMENT: may want the internal formatter to work directly from the message.txt file?

deleting file when done (under user-option control?)

COMMENT: If Moving, delete the file; If Copying, don't.

COMMENT: note that the file is permanent and will not REALLY be deleted, but deleting is a good way to make it empty!

Internal Formatting of mail items

Message Header Property

The header statement of a message will have a message-header property consisting of an inferior tree. This seems nice since it very nicely accomodates any number of header fields. If a field is not present in the message then it will not appear in the header property (I doubt that very many fields in a message will be used in th majority of cases.). A linear tree structure is envisioned in which each node consists of a field containing the text making up that field. This needs more thought:

1. Will this structure allow the user to actually get to it indirectly (it is really simply a plex) by something like Jump <to> Name Any where he happens to guess the name of this statement ?

COMMENT: No. Property lists don't work like that.

2. Some nodes may have a better representation than simple text. Perhaps, each node should be a pair of statements: one containing the field name, the other containing the value of that field.

COMMENT: The field names should be MAIL subsystem global, non-EXTERNAL, STRING CONSTANTS. There's no need to repeat that information for every message in the world. Besides it would make it easier to change the text of the fieldName. Note that each field-value/node has to be identified as to type (presumably in the ring-element somehow), i.e, which field it is the value for. Then you can put anything you like into the SDB or

some field-values may fit in the ring-element.

3. The algorithm that will search this structure for a particular field needs to be considered.

COMMENT: Merely search thru the inferiour tree plex until finding a node with the appropriate type.

COMMENT: If possible avoid having one ring element for each field, else stored mail will soak up ring elements too fast! Isn't it possible to just link the properties off the same ring element?

visible text formatting

The formatter for the header of the message is envissioned as a procedure whose arguments are the property containing all of the header information and a format. The format will be either the default format or a user defined format. Thus, this procedure may do both the original formatting of the message and reformatting of messages by a reformatting command (?) in Mail.

COMMENT: See comment about "defaults" below.

COMMENT: Yes. I'll add a Reformat command to the command-syntax.

inserting text (body) into structure

do well in non-MAIL created documents

restore structure/ signatures for MAIL created documents

COMMENT: and levels

COMMENT on that comment: "restore structure" is hearby defined to mean statements, statement breaks, and all level information.

omit body if greater in size than user-option set limit

Areas to Consider and Questions

default

COMMENT: Choosing what is to be the "default" is more properly the business of Client Services. Note that a "default" will almost always be just the default setting of a UserOption and would thus be meaningless in an internal design. The actual PROCEDUREs should get actual arguments, not some special value which means "do the default thing". The actual argument would be the "default" only in the sense that it came out of UserOptions.

user controlled method

COMMENT: I don't think this should be a concern in an internal design document. Users control things by choice of commands, selections, and/or UserOptions (and sometimes "user-programs").

COMMENT: What needs to be considered is in what form the formatter get the format information, and how it does its thing.

Read Algorithm

```
read <= [[user]

   rdusermail[ getmessages[user]; transfertype [user];
   msgtype[user]]] ]


rdusermail <= [messages, transtype, msgtyp]

   [ null[messages] -> terminate[];

   isdifferent[msgtyp; gettype[first[messages]]] ->

      rdusermail[ rest[messages]; transtype; msgtyp];


   t ->

      rdmessage[ first[messages];transtype; getdest[]],

      rdusermail[ rest[messages]; transtype; msgtyp]  ]


rdmessage <=  [message, transtype, dest]

   [ iscopy[transtype] -> copy[message; dest];

   ismove[transtype] -> move[message; dest];

   t -> "transfer type not implemented" ]


copy <= [message, dest]

   [isfrommail[message] ->

      copymail[getheader[message], getbody[message], dest];

   isfromnonmail[message] ->

      copynonmail[getheader[message], getbody[message], dest] ]

copymail <= [msgheader, msgbody]
```

Previous Thoughts and Misc Junk

COMMENT: Message.txt will be Moved/Copied into a "category". Some categories are "indirect categories", which means that the messages may be stored in some file other than the initials file.

COMMENT: I (DIA) think the message reader should ONLY put the mail in the "new-mail" branch, whether it has been seen in MSG or not. And the names should be something like Mn where n is a sequence number. I think this kind of sequence number should only appear in this branch and be replace by the message-id when moved to a category. However, we should walk thru how a typewriter person is going to read mail and reference items before we are sure of the names.

mailer design

COMMENT: This can probably be ignored until after EOP benchmarks.

needs to handle "whole" files

needs to get across Tymnet

needs to forward mail from "gateway" ARPANET machine to Tymnet machine.

EOP Considerations

Encoding signatures, structure or any other properties in a sequential file only to be decoded at the other end is bullshit! I have struggled with this concept for several days and consider it a waste of time. Mail sent from one Mail user to another should go directly to its destination with no sequential monkey-business. Until an AUGMENT mailer can be written I suggest that the current mailer be used to send only the header of a message containing a Journal citation and let the read routine expand or not as it sees fit. For EOP simply record everything! if you don't like recording everything then sequentialize the message body and don't worry about how it comes out the other end.

If you read <maildev,mailsupport,> in the near future you may need a guide since it reflects the schizophrenic personality that this design project has given me: design for EOP, design for the eventual super-Mail, design a new Mailer, don't worry about designing Mailer because we won't have time to do it!!!!!!

The design for outgoing mail was done with a new mailer in mind. The design for incoming mail (reading) has essentially no changes from my original attempt and strongly reflects the sequential monkey business. Read the Misc Junk branch at your own risk. I think we need to talk about this stuff and settle on something.

P.S. I'm confident that all this struggling is converging to something reasonable.

AUGMENT mailer

Outgoing mail

1. Create Message-ID and Message-ID field.

2. Get Date and create Date field.

3. Create an AUGMENT-TO-GO file to contain the message to be sent. It will contain the complete header minus the bcc field and body. The header properties will be added to the (a?) header statement. The bcc property will not be added at this time.

4. Get address lists of people who will receive the mail item.

   Copy to and cc addresses into a single global string (call it tostring). Copy bcc addresses into another string (call it bccstring). These strings will be used in the actual sending of the mail item.

5. If the message is to be recorded, journalize it immediately.

6. Divide the tostring into Mail and non-Mail addressees.

7. For each Mail user simply FTP the file to his resident host.

8. For each Non-Mail user create an --UNSENT-- sequential file.

   All idents appearing in address fields must be arpanetized.

9. For each individual in the bcc field do the following:

   a) Insert bcc field in header and properties of the header.

   REMEMBER: No bcc individual must know about other bcc individuals.

10. Poke the current mailer.

After Thought:

The sender could control whether the message went through message.txt or not by his address specification: if I give the address as KEV it would not go through message.txt, but if I give the address victor it would .

Incoming mail

Mail comes from two places: 1) the message.txt file and as a result of a direct FTP process of AJGMENT mail.

When an item arrives as an AUGMENT file an attempt is made to insert this item in the user's new mail branch. At this time the user can be informed that he has new mail. If the attempt fails the message is moved into a DELIVERY file. Each user may

have such a file in his directory or a system file could contain all messages waiting to be delivered. The former sounds better to me since it is localized.

To read his mail he simply enters the Mail subsystem which invokes the read routine. The read routine simply copies (and formats the header according to the receiver's user options) his mail from the DELIVERY file into his new mail branch and checks his message.txt file for any mail that will be transferred and formatted.

Good Points about this design

1. This avoids sequentializing an AUGMENT file only to AUGMENTize it at the receiving end. All properties are preserved.

2. We do not have to encode and decode anything.

3. The header properties are inserted once only (at send time) rather than doing it at read time for each copy of the received message.

4. Reading mail from an AUGMENT user is much faster since is simply a matter of copying and formatting.

5. The implementation for EOP would not involve cross host delivery . It would simply involve moving AUGMENT structures around.

6. This design is much cleaner.

Objectionable Points about this design

1. One is not able to see all his mail using only msg.

2. An AUGMENT user gets his mail from two sources.

3. It requires we write a process for the receiving end to handle mail coming from another host. The process would be pretty simple.

4. More code needs to be written. I am not sure how much.

5. We are acting in parallel with the rest of the arpanet community rather than using the same tools.

Statement signatures and any other propertes of statements cannot be encoded into the message without causing irritation to Mail users viewing their mail via msg. I am not convinced that we should make using msg by Mail users more difficult or annoying at this time. Recorded items will have that information in the Journal if a user wanted to see it. Unrecorded mail is usually of a type in which statement signatures are not of interest anyway.

ISSUE: I think it appropriate to add at least one additional field

to the header of a message being sent to a Mail user at send time. When a message is read from message.txt , the presence of this field indicates that it came from a mail user and restructuring the message can proceed immediately. It will of course be thrown away by the read procedure. There may or may not be any useful information stored in this field. For example, it could be used to store restructuring information.

Address control

A MAIL user will input addresses of the following forms:

1. individual ident

2. group ident

   The user should not have to enter all the individual names but they must be filled out at some point. At message creation time or formatting (when send process starts) time?

   Whether or not the Mail receiver sees all the individual names should be under user option control.

3. ARPANET address - nothing to do here

4. Postal mail address - nothing to do here

Determine if going to MAIL or non-MAIL user for each outgoing copy

if non-MAIL must change idents to addresses.

Do this by checking ident file.

COMMENT: I think this must ALWAYS be done to allow even MAIL-users to sometimes use MSG, Hermes, etc. to process their mail.

COMMENT: The only concern here is that a MAIL user may want to answer a message with a non-MAIL program. This is really a user interface issue. I vote for including idents so that I don't have to see user names and host names (and forcing MAIL users to always answer with MAIL). Note that if we go with user names, we are pratically turning the ident system into program reference document rather than something users really use.

return address may be ARPANET "gateway" machine to get to Tymnet

determine if should include body or not (depends on size, addressee)

change Bcc: field as appropriate for addressee.

Standard format for structure to MAIL can get structure back

including signatures.

Make two copies of message. One copy to Mail users will contain structure and signature encoding. The other won't.

May send whole NLS file to the journal rather than sequentializing it

COMMENT: Sug: Don't do this anytime soon as it requires more code to be written and all you get is more effieciency, not more funtionality.

Not for distribution

NLS/Augment Mail

ROM JRP 20-Jun-79 16:50 47646

Review of MAIL design

Overview 1

This constitutes our review of ARC's ongoing design of a new MAIL system. It is based on reading some of the design documents (snapshot taken on 1-Jun-79). Specifically we looked at OVERALL-DESIGN, CATALOG, DELIVERY-MECHANISM, and, MAILSUPPORT (including the comments therein). 1a

While this document may have a flavor of criticizing the design, this is certainly not the case. The general approach taken is innovative and far reaching, but since this is not a detailed evaluation of the design it does not include a list of everything that is done right. It includes some observations of areas that, to our opinion, require more attention, re-thinking, or a more comprehensive approach. We have interesting solutions to some of the problems cited which we will gladly discuss but which we did not have the time to put in writing. 1b

Conspicuously missing from the design is a firm framework and a model that can be derived from such a framework. A most important aspect of a framework are the design constraints which are not mentioned anywhere. This is manifested by differing approaches and assumptions found in the various design documents and causes divergence, rather than convergence, towards the desired system. 1c

The lack of the clear model is further evidenced by the following general problem areas: 1d

Confused Goals. 1d1

The goals of the system and its final capabilities are not specified (a user interface is by no means a way of specifying system capabilities). A distinction must be made between the goals and the means of achieving them; this in particular is vague--milestones, benchmarks, and temporary solutions are all mixed whereas they should be clearly identified. 1d1a

Missing Specifications. 1d2

Many functions are alluded to in the various documents but are never mentioned beyond the desire to have them included. Many other functions are not specified at all nor are they specifically rejected. (Some functions are only partially specified; we attribute this to the temporary nature of the documents.) 1d2a

Intermingling Of Functions. 1d3

Review of MAIL design

The partial specification and the unclear goals caused a poor separation of functions. Implementation consideration are dominant in the functional specification resulting in implementation driven functional definition rather than the other way round. (This for itself makes the vague model even vaguer.) 1d3a

The lack of a comprehensive model is the main finding of our review, and we comment on this some more in the next section. The rest of the sections present our detailed observations. 1e

Framework and Model 2

A general framework can be defined for AUGMENT Mail which defines the application, its goals, and its constraints in a top-down fashion. The overall design would be enhanced by such a framework, and some of the problems evident in the current design would be avoided. 2a

The absence of a general, modularized framework is evidenced by the following problems in the current design: 2b

Journalizing and transmission functions are intermingled 2c

Although there are separate sections in the design document to discuss mail delivery and journal entry, the distinction between the two functions needs more clarification. Mail items and Journal items are similar in some ways but different in others, but the comparison is not clear. Also, the division of responsibility between the journalizing function and the transmission function, especially in the processing of distribution lists, needs to be outlined. 2c1

User Interface and Journalizing/Transmission functions are not tied together 2d

The relationship between the user interface process and the mail delivery process needs more definition. 2d1

A feature that is visibly missing is that of monitoring and controlling the delivery process from the user interface process, e.g. repairing misaddressed mail, altering delivery priority, checking delivery queues, etc. Some fields are needed for the delivery and journalizing functions, while others are not. The roles of these fields, and the methods for passing them out of the user interface environment are attributes which need refinement. 2d2

Arpanet mail model dominates design of AUGMENT Mail 2e

Review of MAIL design

The functions and features of the AUGMENT mail environment are unduly constrained by the desire to remain compatible with the Arpanet mail environment. Two symptoms of this problem are the definition of the user interface in terms of the Arpanet message standard and the attempt to use existing Arpanet mail tools, such as the mailer, in support of AUGMENT Mail. Interconnection with Arpanet mail can be achieved in a way that preserves the clean functionality of AUGMENT mail. 2e1

The existence of a general model of the mail application can help both to eliminate some of the design deficiencies and to more clearly mark the goals and constraints built into the design. With such a model, the interoperability issues with non-AUGMENT environments will be easier to discern. A benefit, which may become important in a short time, is the potential for influencing the national/international communities (such as IFIP WG 6.5) to incorporate functions in the standard models of computer mail which are compatible with AUGMENT Mail. 2f

Journal problems 3

In the current set of design documents the term Journal is not well defined. It seems that at times the Journal, the catalog, and the delivery mechanisms are all called "the journal". This confusing terminology needs to be clarified. Specifically the difference between the MAIL system and the Journal system needs clarification (at least for ARCers). 3a

In the (multi) Journal system specification we find the following items that need more attention: 3b

Messages, headers, and catalog entries 3c

There is no definition of what information belongs in each of these data structures (most notably the message header and the catalog). While information can be used for many purposes it must be clear where it belongs so it is not duplicated unnecessarily. For example, it is still unclear whether a message header needs to be retained after recording or whether the catalog record suffices. 3c1

Message Numbering 3d

This is specifically identified (in the documents we reviewed) as a problem area and hence we will not comment in length here. The issues of unique numbering, cross-journal numbering, and resolving (cross) Journal links clearly need some work. 3d1

Delivery Mechanism 3e

Message delivery is handled bits and pieces everywhere, mostly dominated by issues of ARPANET interface and ARPANET mailer. This has led to discarding some valuable delivery modes (e.g., a citation delivered before the message is ready) and to inefficiencies in general Journal communication. 3e1

A general delivery mechanism need be defined that will support delivery to Augment as well as non-Augment mail systems and that deploys many transmission media. However, this mechanism can be specified only after all desired capabilities are well defined. (In a separate section below we comment on the Augment to non-Augment mail interface.) 3e2

Multi-Journal issues 3f

Supporting a multi-journal is identified as a goal but never really discussed. The list of important relevant issues is long; to name a few: how does one journal communicate with another? how is catalog information passed? how is the the "cache" of foreign journal items stored locally? how is a "foreign" Journal item retrieved? 3f1

These aspects cannot be commented on in a few lines and are clearly non-trivial design issues. It is important to have a good clear interface among journals specified beforehand so that the transition from a single to a multiple Journal system is smooth. 3f2

The Catalog facility 3g

The confusion between messages and catalogs partially stems from the lack of definition of the catalog. The catalog is a data base that must provide at least two functions: search capabilities by catalog entries (author, dates, etc) and address resolution (converting a Journal identifier to a regular Augment link). Aside from identifying the entries in this data base (both in content and form) its general functionality must be specified. Some examples of issues that must (and currently are not) addressed are: how it is accessed and by what processes, how are catalog entries communicated within a Journal system and cross Journals, how are catalog functions of a foreign Journal performed, etc. 3g1

Ident System problems 4

Addressing in any mail system requires careful design and exact definition but received only cursory attention in the current design. In the MAIL system addressing is based on a (multitude of) ident systems which themselves are not clearly defined or

specified. At least that portion of the ident system that is used by the MAIL system must be exactly specified. 4a

In addition to MAIL the ident system serves other AUGMENT related functions (e.g. statement signatures) and the transition from a single ident system to multiple systems may not be straightforward. It is mandatory to explore the interrelations between the various entities that use ident systems so that adverse side affects are avoided. 4b

With regard to the MAIL system the following issues need re-evaluation: 4c

Individual membership in an ident system. 4c1

It may be desired that individuals belong to more than one ident system since they should be allowed belong to more than one AUGMENT "system" (such as the architects, in their own system and the ARC system). Of course one can include the same individual in different ident systems under different idents, but that is cumbersome (statement signatures, again). 4c1a

Group idents across ident systems 4c2

The advantage of group idents will be lost if membership cannot span more than one ident system. Related issues such as to which ident system does such a group belong are not discussed at all. Group idents that include other group idents will, of course, complicate the problem. 4c2a

Relation between Journal systems and Ident systems 4c3

One can conceive of many relations between Journal system and ident systems which must all be explored. The simplest and most restricted is a one-to-one relation where each Journal system has its "own" ident system and vice versa. Another possible relation is a single ident system for many Journal systems. Many other possibilities should be explored including a structured ident system that has other ident systems included in it. 4c3a

Address resolution 4c4

Resolving an ident to the individuals' mail boxes is obviously an ident system function. The current design does not specify how this is done. Issues such as where are idents expanded (especially group idents), how and when are

"foreign" idents resolved, and what are the ident validation facilities must all be discussed. 4c4a

Non AUGMENT ident systems 4c5

While a lengthy discussion of interfacing the AUGMENT MAIL system to other mail systems (specifically the ARPANET) is presented there is no specific discussion of interfacing to other, non-AUGMENT, ident systems. Such systems do exist and although may not be called ident systems they perform similar functions (at least in the MAIL context). 4c5a

It is our feeling that the ident system is, and should be, kept as a separate entity as it performs functions that are not unique to the MAIL facility. Yet even as a separate entity it should be well integrated into the model on which the MAIL system is based. 4d

AUGMENT to non-AUGMENT interoperability 5

Mail systems in dissimilar environments will never be completely compatible with each other. International standards will foster compatibility to a degree greater than seen today, but each individual mail system will support unique functions which are not available outside of the local environment. 5a

The most complex non-AUGMENT mail system currently known, Arpanet mail, is but one example of an external mail systems to which AUGMENT may require connection. The functionality inherent in Arpanet mail should not unduly constrain the design of AUGMENT mail, especially since AUGMENT mail is much richer in the support of shared data base message exchanges. Likewise, no attempt should be made to utilize the Arpanet mail tools within the AUGMENT environment, or vice versa. 5b

Rather than recommend the specific approach to be taken, we recommend a reevaluation of the current approach. If a goal exists to interconnect with other, not-Arpa, or the next generation of Arpa mail systems, then greater care is needed to identify and generalize the functions performed within AUGMENT mail. Once those functions are encapsulated into a clean functional model, then the task of interconnecting with other mail systems becomes more tractable. 5c

Powerful and clean approaches to this problem of mail system interconnection do exist. These approaches can and should be adapted to meet the needs of the AUGMENT Mail - Arpanet Mail interconnection. 5d

AVOIDING WORKING NON-SOLUTIONS TO OFFICE COMMUNICATIONS SYSTEM DESIGN

James H. Bair

BNR INC.
Palo Alto, California

SUMMARY

Computer-based message systems are no longer being used exclusively by computer sophisticates. The new users, primarily office workers, bring a totally different perspective to the use of digital technology. A review of lay-user's perspective reveals problems in current approaches to system design that typically result in "working non-solutions". Determining working solutions is not merely a matter of surveying potential users as they represent but one of the four divergent perceptions of the ideal system design. A design solution is proposed that facilitates convergence of the perceptions of the system chooser, user, manufacturer, and researcher. A current research effort is described which predicts, based on current attitudes and behaviors, a system design that potential users will accept and that will enhance productivity in the future.

BACKGROUND

Computer-based message systems have evolved considerably since they appeared as an adjunct to the early development of timesharing well over a decade ago. Initially, programmers adjacent to R&D computer centers valued leaving messages for colleagues at about the level of another few lines of good code. Covert evolution of message handling software soon resulted in an awareness of a new medium of communication for white-collar workers.

Network designers and users reminisce about the almost startling discovery that the ARPA Net was being used more for messaging than for data transport and other applications for which it was designed.[1] Experienced computer users were engaged in nationwide conversations irrespective of location or hardware. We are now on the verge of another major proliferation of computer messaging represented by PLANET, ONTYME, COMET, HERMES and other personal mailbox, store and forward message systems. The potential and impact of sophisticated computer messaging as it has evolved since its earliest days is reported in two new books, The Network Nation, by Hiltz & Turoff[2] and The Office of the Future, by Uhlig, Farber, and Bair.[3]

The current proliferation of message systems represents a discontinuity in the preceding evolution. Messaging, along with other direct uses of computers, is leaving the domain of highly trained and skilled computer sophisticates, fluent in the cryptic language of machines, and appearing as a commercial product in realms where computers are more likely to be feared rather than loved. The non-sophisticated computer user has a totally different perceptual and experiential perspective than the sophisticate. This difference will not change the underlying purpose for using messaging products, but it will change the interface and performance characteristics required for acceptance.

Experience with the attempted implementation of messaging systems that did not accommodate the lay user's perspective has generally been negative. Examples abound, but it is more useful to concentrate on the nature of the user's perspective. A potential user's perspective has three fundamental, psychological components: (1) needs, (2) characteristics, and (3) expectations.

(1) User needs include both basic human needs and job requirements. The basic human needs are perhaps best characterized by Abraham Maslow.[4] In essence, Maslow explored the notion that we are motivated by needs, with certain needs having priority over others. If using a system is perceived as either enhancing a user's job security, providing additional socialization opportunities, or contributing to his esteem, in that order, there is positive motivation. If there is no perceived relationship to these needs, motivation for serious usage seems improbable. Motivation to use a system may be intuitive to system designers, but not obvious to computer non-users. Concrete portrayals of system benefits to prospective users are required that relate to job success and socialization needs. It must be remembered that the user's need is for accomplishment of his job with the computer as a means not an end.

(2) User characteristics include the areas of sensation, attention, memory, and cognition. Most system interfaces are depraved as sensory stimuli, consisting of monochromatic, one dimensional CRT's with the resolution of drawings scratched on a sandy beach. Many systems violate memory processes by requiring a user to operationalize more than the average maximum of seven chunks of information retainable in short term memory.[5] Mnemonic aids to remembering commands and operators also seem to be ignored.

CH1491-0/80/0000-0099$00.75

(3) User expectations are probably the most powerful influence upon the user's experience. The user brings expectations derived from use of other technologies and systems, most notably the telephone and typewriter. Overt telephone failure is almost non-existent, with inefficiencies relegated to operator error and caller unavailability -- problems external to the technology. Typewriters do fail, but are easily replaced, repaired, or ignored. In both cases, the technology behaves in a predictable manner, fitting a model easily learned by the user. Reliability, responsiveness, and performance as advertised, without surprises, are imperative to avoid negative reactions.

To date, unfortunately, we are far from realizing systems designed in keeping with these components of the user's perspective, although such designs have been described.[6] At the least, users experience frustration and stress caused by the behavior of systems that should be as operational as the telephone, but are not. Subsequently, even adventuresome users can reject further usage and skepticism of future technological innovations spreads.

SYSTEM DESIGN PROBLEMS

The Current Design Process

Correction of any situation requires identification of the problem areas; unfortunately, the whole of current design practice appears at fault. But the fault is only apparent if we align ourselves with the user's values and perspective. The traditional design is driven by the highly technical, precise, and demanding computer and engineering sciences. To produce good programmers our educational system begins infusing them with the utmost of empirical rationality at an early age. A new way of thinking and problem solving is inculcated, derived from the abstract tools of mathematics and engineering. At the end of 4-6 years, this educational process has forever engendered a new thought process which is brought to bear on design. Collectively, a culture with its own language is created -- the rigorous, logical, domain of programming languages and input-output devices.

In this culture, the questions posed do not arise from user needs for the most part, but rather from technology potential: "What neat thing can I get my software to do?" The result of competent effort is a system that functions well for the programmer-designer. Of course, accommodating the user is merely a matter of adding some user support -- documentation, training, and an introduction to computers for good measure. The end result typifies message systems and other office systems today: a working non-solution.

The resultant system typically can be demonstrated and shown to perform in some way, for example, delivering and managing messages. However, the lay-user is confronted with operational opacity, undecipherable error conditions and error messages, and a plethora of control characters, function keys, and arbitrary steps that must be followed unconditionally. The learning process threatens the user's ability to meet basic needs (including job requirements) by consuming time for learning and operation. This same time loss has negative bottom line impact. To meet the basic needs of users, a system must be transparent to operate, minimizing time loss, and consistent with performance expectations.

Determining User Needs -- Divergent Perceptions

Perhaps design problems would be readily solvable if the solution were merely to meet "user needs". However, there are four very different perceptions that confound even the most user-oriented design approach: (1) what the system chooser will buy, (2) what the user says he needs, (3) what industry can develop, and (4) what will actually increase user effectiveness.

(1) The system chooser is the buyer or decision maker in an institution acquiring a service. His criteria are primarily economic, including a limited investment per person, a predetermined rate of return, and a defined amortization period. Very often, up-front economic constraints prohibit investment in design and support that are fundamental to achieving critical mass (e.g. having enough users) and long range viability.

(2) A classic misconception is that the user can tell designers what he needs. No different than life in general, the process of self-diagnosis is woefully inadequate as the sole determinant of needs. Certainly, users must be polled and seriously regarded in selecting opportunities for system application, but super-human powers of prediction should be left to others. A user cannot anticipate the form of a technology totally unknown to him; energy is much better spent determining the user's values and criteria for success in the context of his current work environment.

(3) Even if the user could anticipate the optimum design of useful technology, industry currently would have difficulty delivering the product. In numerous consulting studies, the author has found that currently available systems could not meet the chooser's or the user's criteria. Perhaps we were seeing industry's reaction to user demands which resulted in a situation where neither user requirements or industry capability were met, almost a "Catch 22" effect. Industry, dominated by the working non-solution approach, perceives users' needs in light of the current capabilities of mass-produced technology. For example, if the user wants an integrated telecommunications system for voice and data, he most likely will get a telephone interconnect switch that requires parallel wiring for any digital transmission with users. Or, for office automation, he likely will get a word processor which uses special purpose hardware for mechanized typing in total disregard for actual office functions; in other words, a mass produced non-solution. The limitation is not the potential of industry, but industry's perception of system design, an idea vividly presented by Morton, et al.[7]

(4) The convergence of the foregoing three perspectives does not yet address the most important question: What will increase user effectiveness? Plagued with two subproblems, how to measure effectiveness changes and how to determine the causal relationships to design variables, this perspective remains a frontier for design related research. In an overview of a measurement methodology for electronic office systems,[8] the author proposes a conceptual model of the variables that must be measured, methods of measurement, and the measurement controls that must be followed. Some of the results of measurement, particularly the causal relationships for improved efficiency are reported in other publications.[9, 10] A surprising conclusion from this work is that users cannot judge what will improve their effectiveness. Measurement of different design factors must be done indirectly, regardless of whether the factors are cursor control or the functional capabilities of a message system.

For example, users generally appear to resist buying or using the "mouse" cursor control device. However, years of research and use at Stanford Research Institute[11] and elsewhere[12] have shown very significant improvements in the efficiency of editing and other interactive CRT tasks using the mouse. The improvement is so great that it appears worth any effort to add the mouse to present workstations or obtain a license. A second example is the difference between initial user perceptions of required functional capabilities of message systems and their perceptions after several months of experience. Invariably, the desire for a simple system with few commands for sending short messages gives way to dissatisfaction with the restricted service and a demand for extended capabilities such as document production.

In both examples users could not explicitly define the interface design or the capabilities and services they require to increase their effectiveness. Other examples show the limitations of user's subjective assessments of their needs. Observational studies of the time spent in daily office activities continually surprise subjects who subjectively anticipated far different results. Two findings were notably unbelievable by users: 75% of managerial time is spent in communication activities,[13] and 20% of clerical time is spent waiting for work.[14]

Thus, user's perceptions must be taken as only one of four perspectives of system design, and do not reflect valid changes in effectiveness. However, the user's perspective as stated above is a very important component in design success. User's perceptions can provide an understanding of expectations of system characteristics, and provide the basis of user participation in design, both critical for successful system implementation and acceptance.

A SYSTEM DESIGN SOLUTION

Design from a Model of the End-User

The design process must incorporate the four perspectives into a working solution that begins with measuring and analyzing the behavior of users, choosers, and designers, and then overlaying the results on technological feasibility. There are basically seven steps leading to a prototype system.

1. Survey User Expectations. Determine the subjective anticipation of pre-users and characterize the potential for accepting technological innovation in their work life.

2. Measure User Needs. Apply work flow analysis, work measurement techniques and psychological instruments to representative office operations and laboratory simulations to gather data about what actually will impact user effectiveness.[8, 15]

3. Use Psycho-social Characteristics. Use the literature on human performance and laboratory studies of human-computer systems as guidelines for specific design features such as cursor control, and command language consistency.[16]

4. Establish Functional Requirements. Map a progression of functional capabilities from initial implementation through system maturity for introduction to users; for example, a simple messaging capability could be followed by a personal management information system.

5. Use Efficiency Data. Design the interface, and work methods and procedures based on time and motion like studies of simulated human-computer communication scenarios.

6. Specify Systems Features. Specify each feature based on weighted valuations from efficiency and effectiveness results, psycho-social characteristics, and user acceptance potential.

7. Overlay Functional & Feature Specification on Technology. Carefully revise the tentative design into feasible packages to be introduced incrementally as technological development permits. This emphasizes deliverables that are reliable, responsive, packaged into non-intrusive hardware, and when they can be delivered.

This brief description of the seven steps is intended to draw attention to the user behavior-driven design and the need for thorough end-user research prior to any development of message or other systems intended for non-computer sophisticates.

An End-User Research Methodology

Our present work focuses on an approach to the front-end of the design process (Steps 1 & 2) and is oriented toward the perspective of the user and chooser as described above. (Previous work has focused on the perspectives of industry and end-

user effectiveness.) Our methodology is a comprehensive package of research methods that describes the potential for usage as well as design factors for office communication technology in an organization. As a methodology, it provides all the necessary information to conduct an on-site study. For example, the methodology includes the step-by-step procedures for administering questionnaires, introducing the study to respondents, and the reasons why each question or measurement is used. To develop an accurate base for the design process, several different research methods are used that complement each other in a coordinated package.

The methodology is intended to answer two questions: what technology will enable users to be more productive (perspective 4), and what will be accepted in the working environment (perspective 2)? No single research method will address both these questions. The productivity component requires quantitative data and a model of what improves productivity; the acceptance component relies more on subjective data. Therefore, the methodology includes both quantitative and subjective types of research methods.

The research methods have been designed with the most challenging question in mind:

> How can we predict what potential users will accept that will make them more productive in the future based on current attitudes and behaviors?

The methodology addresses this question by using research tools that do not ask a respondent questions about technology of the future, or about other things which the respondent has not experienced. All questions are in terms of past experiences or attitudes currently held.

Likewise, measurements are of current behaviors from which future behaviors can be predicted. Thus, the prediction process is done by the research team, not the respondents. A complex model (conceptual) based on years of extensive investigation of prototype office-of-the-future implementations[3] enables the best possible prediction. Basically, this works by comparing the new data to previous data. Where there are significant similarities between the data, we can predict similar levels of acceptance and effects on productivity between the current study population and previous study results.

The data gathering process in each case begins with the selection of a company representative of a market segment; for example, electronics manufacturing. After an understanding of the company is obtained from its management, a representative sample of about 200 persons is selected. The sample consists of offices where the primary activity is information processing. A set of questionnaires is given to each person. Based on the questionnaire results, 20 - 40 respondents are interviewed. A final rating questionnaire is given to all respondents which focuses on the technological improvements possible in the office. An optional method, the communication audit,[17] may then be used to quantitatively describe communication behaviors such as meetings, telephone calls, and memos. During the process of delivering and collecting questionnaires and interviewing, observations are made of information management procedures, working environment, and working methods.

Upon completion of the case study, the data is summarized and analyzed showing the correlation between the human, organizational, and design variables. The results will identify the opportunities to use advanced office systems and the acceptance level of the potential technological solutions. "Opportunities" for use reflect user needs without relying upon the user's judgment. Knowing the "acceptance level" permits selection of the appropriate functions, features, and configuration for a target implementation date. This approach, very briefly desribed, facilitates the successful convergence of the design goals of the chooser, user, and industry, and the requirements for improved efficiency and effectiveness.

In conclusion, we note that the reason for being for computer communication systems is to serve users who are no longer the designers! And to quote Morton, et al.[7] "What we know above all is that the new user is most emphatically not made in the image of the designer."

REFERENCES

1. Lukasik, S.J. "Organizational and Social Impact of a Personal Message Service," Proceedings of the National Telecommunications Conference, IEEE Communications Society, San Diego, Calif., December 1974.

2. Hiltz, R.S., and M. Turoff, The Network Nation, Human Communication via Computer, Reading, Mass.: Addison-Wesley, 1978.

3. Uhlig, R., D. Farber, J.H. Bair, The Office-of-the-Future: Communications and Computers, Amsterdam: North-Holland Publishers, 1979.

4. Maslow, A.H. Toward a Psychology of Being, Princeton, N.J.: Van Nostrand, 1962.

5. Bair, J.H. "Human Information Processing in Man-Computer Systems," Paper presented at the annual conference of the International Communication Association, Phoenix, Arizona, 1971. (Information Sciences Abstracts, Vol. 7, No. 2, 1972).

6. Watson, R.W. "User Interface Design Issues for a Large Interactive System," AFIPS Conference Proceedings, National Computer Conference, Vol. 45, 1976, pp. 357-364.

7. Morton, J., et al. "Interacting With the Computer: a Framework," in Teleinformatics 1979, IFIP International Conference, Amsterdam: North-Holland Publishers, 1979, pp. 201-208.

8. Bair, J.H. "Productivity Assessment of Office Information Systems," Proceedings of the IEEE Symposium on Trends and Applications in Distributed Processing, National Bureau of Standards, May 1978.

9. Bair, J.H. "Communication in the Office-of-the-Future: Where the Real Payoff May Be," Proceedings of the International Computer Communications Conference, Kyoto, Japan, September 1978.

10. Bair, J.H. "A Communications Perspective for Identifying Office Automation Payoffs," Invited Paper at the New York University Symposium on Automated Office Systems, New York, May 1979.

11. Engelbart, D.C. "Design Considerations for Knowledge Workshop Terminals," AFIPS Conference Proceedings, 1973 NCC, Vol. 42, pp. 221-227.

12. Card, S.K., W.K. English, B.J. Burr, "Evaluation of Mouse, Rate-Controlled Isometric Joystick, Step Keys, and Text Keys for Text Selection on a CRT," Ergonomics, Vol. 21 No. 8, 1978, pp. 601-613.

13. Mintzberg, Henry. The Nature of Managerial Work, New York: Harper and Row, 1973.

14. Purchase, Alan. "Office of the Future," SRI International Business Intelligence Program, Guideline No. 1001, 1976.

15. Tapscott, D. "Towards a Methodology for Office Information Communications Systems Research," Paper presented to the International Workshop on Integrated Office Systems, IRIA, Versailles, France, Nov. 6-9, 1979.

16. Bair, J.H. "Design Problems and Guidelines for Human-Computer Communication in Office Automation Systems," Invited Paper, Proceedings of EuroIFIP 1979: European Conference on Applied Information Technology, London, England, Sept. 1979.

17. Conrath, D.W., J.H. Bair, "The Computer as an Interpersonal Communication Device: A Study of Augmentation Technology and its Apparent Impact on Organizational Communication," Proceedings of the Second International Conference on Computer Communications, Stockholm, Sweden, August 1974, pp. 121-128.

Don M. Avedon
Automated Datatron, Inc.

the Automated Office

Electronic Mail Systems

Electronic Mail provides an exciting and revolutionary new approach to communications in today's fast-moving business environment. These systems can manage virtually every aspect of person-to-person communication within an organization, including message preparation, transmission, filing, retrieval and distribution control. Most of these systems use a computer as an electronic mailbox and give each user access to his or her messages through ordinary computer terminals — fixed or portable, video or hardcopy.

In order to see the advantages of Electronic Mail, let's compare it with conventional forms of business communications:

• Unlike an office memo, the computer-recorded message cannot be lost or misplaced. The system gives both sender and receiver time-stamped proof of transmission and a record of the message.

• Unlike the telephone, Electronic Mail lets you send a message without knowing where the recipient is. All messages go to the central computer and can be picked up from any terminal on the system. Because the computer holds the messages until they are called for, there is no need for both parties to be available at the same time.

• Unlike intra-office mail or the postal system, Electronic Mail is instantaneous. It lets you compose, edit, transmit, distribute and file a message in a fraction of the time it would take to send a business letter — minutes, instead of days.

• Electronic Mail expands your communications "window." When you call by phone, within the same time zone, you have just 40 hours a week to make contact. And if you are calling cross-country, your contact time is cut to 25 hours. With Electronic Mail you can send and receive messages whenever you want, without worrying about time-of-day. Your contact time is a full 168 hours a week.

• Electronic Mail Systems are totally non-interruptive. They don't ring, buzz or break in on meetings. You use it only when it is convenient for you.

The following are the main applications:

1. **Broadcasting** — to send the same message to multiple devices. Example: Have a memo sent to ten people via one command.

2. **Store/Carry Forward** — the store option would automatically keep messages available for a defined period of time, if the receiving device was not available. The carry forward option would give individuals the capability of redirecting messages to different areas. The combined functions would store messages until a user becomes available anywhere on the system.

3. **Calendaring** — does what its name indicates. It keeps track of people, places and time.

4. **Feedback/Response** — the system automatically tracks and confirms delivery of messages. In addition, the receiving party may respond.

5. **A Total Network** — a network system will not only allow local devices to communicate with one another, but also allow all functions and information to be available to remote sites.

The output the user receives may be soft copy on a display, hardcopy produced by high speed page printers or letter quality printers and/or microfilm/microfiche from computer-output recorders.

Simplicity for the User

These systems are designed from the users point of view; the computer does most of the work and helps the user. You do not have to be a programmer nor even know how to type to use electronic mail. You talk to the computer in plain English. Here is a list of typical functions and a brief explanation of each:

Compose: You just create as you normally do, in your own words, in your own style. Anything from a one sentence note, a few sentence memo to a multiple page document, may be created.

Edit: The computer makes it simple to make changes on the spot. You can back up, correct mistakes and make additions or deletions until everything is just the way you want it.

Send: The system automatically transmits your message to the people you name — one person or many. You can indicate a distribution list, such as "Marketing Department" or "All Managers" and automatically send the same message to each person on the list.

Continued on page 66

As Vice President of Automated Datatron, Inc., Don M. Avedon is responsible for marketing services, new business strategies and sales activities. He is a Fellow of the National Micrographics Association (NMA) and a former member of the NMA Board of Directors.

Avedon was previously vice president, Micronet, Inc.; vice president, research and development, Microfilming Corporation of America; technical director, NMA; and project engineer, Bell Telephone Laboratories. He authored the book Computer Output Microfilm *and was formerly the editor of the NMA's* Glossary of Micrographics *and* The Journal of Micrographics.

Penn/from 10

sult was that the importance and viability of any records management program became directly related to the personality of the individual records manager in charge and had nothing whatever to do with the program's functional necessity. It remains so to this day.

Will information managers fare any better? I don't think so. If he is to do his job properly an information manager is going to have to ask the program manager some pretty tough questions about his information. The program manager will go along for a while. He might initially even be supportive. But ultimately he will become frustrated by the interference and a head-to-head conflict will ensue. The program manager will prevail. Why? Because no matter what sort of fancy title the information manager has and no matter what his position description says, the fact remains, *that he will be staff — and the program manager will be line. When line argues with staff, line wins. Line is why you're in business; staff is there for support.*

"No matter what fancy title the information manager has and no matter what his position description says, the fact remains that he will be staff and the program manager will be line."

If top managers are serious about managing their information (and regardless of the rhetoric I'm not yet convinced they are) then the long dead information analysis and control functions of records management should be resurrected and reestablished as a part of the records management program. Considering past experience there is no guarantee that this action will result in permanent improvement, but at least there is a chance that it will do so. There is, after all, a structure to build on. If it is necessary for "political" purposes to call the "new" program Records *and Information* Management, so be it. It is a redundancy, but it's a small price to pay and no worse than records and forms, or correspondence and records, or records and files, none of which make much sense, but all of which we've seen. Of course, if top managers are not serious about managing their information they can continue to virtually ignore the records management program and treat information management as a separate operation.

Currently, the handwriting on the wall indicates that top managers are leaning toward establishing information management programs with records management as a sub-element. While that is a "tail wagging the dog" arrangement, it is preferable to separation. If such a situation occurs records managers should simply keep in mind that often, to reach a goal, one must maneuver in what appears to be a ridiculous manner. In a sailboat, for example, you must go sideways to go forward. Becoming a part of an information management program may well mean going sideways. Fighting the winds and not becoming a part, however, may well mean going under.

Am I suggesting surrender? No! I am recommending pragmatism. We've been through all this before, you know. Every few years there's a new scheme to save the world — a scheme that is separate from and wants no part of records management. We've had micrographics, which was "not records management" but is now accepted as a records management technology useful for records maintenance and lately, with COM, for records creation as well. We've had word processing, which was "not records management" but which is now seen as a records management technology useful for records creation (and records disposition if the switch is turned off at the wrong time). And now we've got information management which is "not records management" but nevertheless contains the heart of what records management has always been about and which is suddenly touted as a new discovery. Give it time. Things will end up where they're supposed to be. No matter how often you shake the milk the cream always rises to the top. □

"When line argues with staff, line wins. Line is why you are in business; staff is there for support . . . the answer is pragmatism — not surrender."

Avedon/from 12

Scan: In most systems, this function will allow you to get an overview of your electronic files at any time. A one line description will tell you the subject of the message, the date and the name of the originator.

Read: At your convenience, you can have the system display or print out on paper or microfilm, any message or all of them.

Answer: When you want to respond to a message you have received, the system automatically does all of the addressing for you and even attaches a reference to the original message.

File: These systems can allow you to set up a complete filing system in electronic memory. You can have in electronic memory your chronological file, suspense file as well as subject files.

Retrieve: In seconds, you can obtain any information you've filed. It can be retrieved based on subject, date, originator or unique identification code.

There are many possible variations of Electronic Mail systems. You may use your own computer, time share a computer or have no computer at all. You may have dedicated communication facilities or you may use the voice grade public telephone system. You can also subscribe to a system and not own or lease any facilities. Each organization must determine the type of system that best satisfies their needs.

Electronic Mail systems continue to gain momentum and probably, close to one third of the Fortune 500 companies and many government agencies today, use some kind of Electronic Mail system. Most of these systems, however, are only used internally for time-valued information. In the next two to five years, as satellite communication systems come into wide use, as word processing grows and as the concept of automating the office spreads, the use of Electronic Mail will replace much business first class mail and voice phone calls, and it will be one of the sub-systems to the *total information transfer system,* The Office of the Future.

Abstract:

Productivity Assessment of Office Information Systems Technology*

James H. Bair, Senior Information Scientist
SRI International

For the Symposium on
Trends and Applications in Distributed Processing
National Bureau of Standards, May 18, 1978

This paper addresses the problem of measuring the impact of information system technology on office productivity. There are three reasons why there are few useful studies of this problem area: (1) current office information system (OIS) implementations are not mature enough to provide a valid test bed, (2) the resources have not been allocated for significant test implementations, and (3) a comprehensive evaluation methodology has not been developed. Previous work has established strategies for innovation transfer that can result in mature implementations. Implementations are currently under way that will provide the population size necessary for significance, and we hope to be able to persuade the host organizations to allocate resources for evaluation. The third reason is addressed by the work reported in this paper, the development of a comprehensive evaluation methodology.

The approach to the problem is organized into five areas: (1) establishing the scope and definition of OIS, (2) designing a framework to structure the problem area, (3) listing and defining the variables, measures, and measurement instruments, (4) describing the interrelationships among the variables, and (5) designing measurement plans.

OIS is defined by describing the various activities in the office resulting from the use of technology in place of conventional working tools. Activities are the processes of individual work that are common to the different work tasks. Thus, the conventional task of composing text, e.g., writing a paper, is usually to write longhand or to dictate. The activities of writing or dictating will be changed by OIS. For example, online composition, either through a keyboard or some other input medium, will be the new activity. The task of composition will continue to be necessary, but the activities necessary to perform the function will differ.

Productivity is affected by the performance of many components of an OIS that differ greatly in the scope of

variables that must be taken into account for meaningful measurement. We have organized these components into a four-level framework where performance at each level depends on the performance of the level below it: (1) equipment performance, (2) throughput performance, (3) organizational performance, and (4) institutional performance. Choosing a level for an evaluation will define the variables that must be taken into account and narrow the scope of the evaluation. The definition of levels of performance also provides a framework for organizing the results of previously reported studies. The level of a study will indicate the validity of the results.

The measures listed in the paper are representative of a comprehensive list which would be applied to evaluate a system implementation. They are quantifiable and thus can be used for numeric comparisons. More subjective measures are also described. Each measure is described in detail, and matrices show the hypothesized interdependencies and causal relationships.

Measured changes depend on the manipulation of variables that are listed and described. Each of the causal variables listed includes a sublist of the possible alternative values. Changes in the value of these variables will result in changes in the quantitative measures. In some cases there is a fine line between the different values, for example, the difference between "textual drafts" and "typing reports" under "characteristics of application." However, the alternatives are necessary for a manageable description of the potential causes of measured variation.

The way in which these causal variables change the measurement results is a set of relationships that is a model of office information processing. For example, does the use of "multiple single stations" instead of a "centralized system" change the "job completion rate?" The proposed model is primarily theoretical based on the very limited results currently available. There are few conclusive empirical data about these relationships. We briefly present what is known and refine the outstanding questions as a basis for further research.

Measurement plans that intend to be rigorous, comprehensive, and quantitative must control the numerous extraneous variables. Two measurement plans are proposed, a pretest-posttest design, and the same design augmented by a comparison or "control" group. The measurement setting is a mature implementation of complete OIS for a significant user group in an operational environment. The extraneous variables include Hawthorne effects, maturation, selection, history, individual differences, test effects, and others that are common in field evaluations. The execution of such

a rigorous plan is difficult but necessary for a valid determination of changes in productivity. We propose that the comprehensive approach described here is a feasible and reasonable solution to the problem of OIS evaluation.

*Sponsored in part by the Government Service Agency, National Archives and Records Service

COMMUNICATION IN THE OFFICE OF THE FUTURE:

WHERE THE REAL PAYOFF MAY BE

James H. Bair
SRI International

Submitted to the International Computer Communications Conference, Kyoto, Japan, August 1978

Prepublication Draft 31 Jan 78

COMMUNICATION IN THE OFFICE-OF-THE-FUTURE: WHERE THE REAL PAYOFF MAY BE

James H. Bair
SRI International

Abstract

Office activities will be changed in the office-of-the-future and the area of greatest change may be in communication. An examination of the resources of U.S. business shows that non-clerical personnel are the largest labor costs, and the principal activity of non-clerical staff is communication. Thus, the greatest leverage for the benefits of office automation is in supporting the communication activities of non-clerical personnel. An analysis of the benefits of computer mail based on the labor savings in four sample areas indicates that computer mail is cost effective. If the social problems can be overcome, computer mail could well provide the highest payoff of office automation.

Introduction

As early as 1966, J.C.R. Licklider postulated about a "man-computer" symbiosis where man and computer form a synergistic system much more capable of processing information than either man or computer alone. Douglas Engelbart pioneered the development of such a system to extend human intellectual and interpersonal communication abilities, beginning in 1962 and continuing to the present. Several hundred users have explored and applied the "Engelbart system" since the mid-sixties, and the system and its uses have continued to evolve toward the "office-of-the-future." In parallel, many other human-computer systems have materialized to extend and improve the capabilities of white-collar workers to deal with an ever increasing amount of paper and information. In some cases, less sophisticated hardware, such as "word-processing systems," has shown substantial improvements within a limited application domain. However, measurable improvements in overall organizational productivity have been extremely difficult to document in the white-collar area.

In grappling with the challenge of demonstrating the hypothesized improvements in productivity for several years, I have been confronted with many of the problems. One problem has been translating obvious qualitative benefits into quantitative results. Ideally, it would be possible to show an overall increase in organizational productivity from office automation--an increased return on the investment to support white-collar workers. The elusive nature of office work makes this much more difficult than showing gains from automation in industrial work. A second

problem is the lack of funds to develop measurement methodology that would be adequately rigorous and comprehensive. We currently have some contracts, and are just beginning to formalize long standing ideas and speculations.

This paper represents some of the speculation about increased productivity through human-computer systems in one area, interpersonal communication. There is an assumption made here based on some previous research: computer mediated communication or "computer mail" will be used as a substitute for other modes of communication. Based on that assumption, we are able to show that the quantity of communication as a daily activity for a majority of the labor force provides considerable leverage for improvements. Some of the improvements resulting from the use of computer mail appear to be quantifiable to the degree necessary to demonstrate labor savings and provide an adequate return on investment. This is feasible if we do not find major psychological and sociological factors that would negate potential gains.

A Description of the Office-of-the-Future

Since the context of this discussion is the office-of-the-future, and it is not yet a clearly developed concept, a brief description follows. The office-of-the-future describes a combination of new technology and new activities. The new technology, as far as its use in the office is concerned, will be highly interactive based on large, mixed text and graphics displays. We can predict that there will be hierarchical hardware, including intelligent terminals, local shared logic, and network connected, large scale systems. However, hardware is not critical to the concept of the office-of-the-future, particularly since the technology will continue to evolve.

It is most critical that the technology be user oriented, permitting the non-technologist to gain a high degree of fluency with the system with a reasonable amount of effort. I call this degree of fluency "system transparency," because the user sees through the technology unencumbered by the mechanics of usage and free to deal directly with the information. In other words, the computer based media become as comfortable as conventional media are in the office-of-the-present.

Realization of system transparency on a large scale will probably require the elimination of "programmer think" from system design. The special perspective that programmers have often does not include the needs of the inexperienced user. Programmers think in terms of computer operation, a very logical, discrete, binary like process compared to the average user. The acquired logic is good for cross-word puzzles, "space war" and other games, but it cannot be required that users in the office-of-the-future have the programmer's logic.

In the office-of-the-future, there will be different activities in the office resulting from the use of technology in place of conventional working tools. Activities are what people do that

reflect the way they work. Thus, the conventional method of composing text, e.g., writing a paper, is usually to write longhand or to dictate. The activities of writing or dictating will change in this concept of the future office. For example, online composition, either through a keyboard or some other input medium, will be the new activity. The function of composition will continue to be necessary, but the activities necessary to perform the function will differ.

The primary activities that will change include:

- Composition and drawing will be done through online text editing and graphics facilitites.
- Reading and scanning will be done through online portrayal, selection, retrieval and presentation facilities.
- Document production and control will occur entirely online with little needed for the paper medium.
- Messages, mail, memos, telephone calls, and publication distribution will be done largely through computer communication facilities which may include voice, pictures, and processes as well as text.
- Meetings will occur through teleconferencing as well as face-to-face greatly reducing the need to travel, or perhaps even the need for a central office building.
- Information organization on all levels will be prescribed by the user and maintained by the computer. The relationships between pieces of information will not be static as in a file cabinet, but dynamic so as to represent conditional relations.
- Personal information management will be done online including notes, agendas, calendars, etc., and these will actively remind the user of due dates, meetings, etc.
- Organizational information management will be integrated into the other activities, particularly those of the individual. Communications, reports and papers, financial and production data, legal and planning information and so on will be interlinked so that related pieces of information are tied together forming an organizational intelligence.
- Intellectual and creative activity will be supported through the other activities, but also will be extended by the active computer medium which can maintain relationships among different pieces of information so that more abstract relationships between the relationships can be explored.

In addition to changes in activities, there will be changes in the way in which services are used and accessed. Most services that white collar workers use will be accessed through a single interface. Figure A-1 shows the worker in the office-of-the-future with services that are available to him. The services are grouped into three lobes. The first lobe includes the services that are most personal and everyday--core services. The more

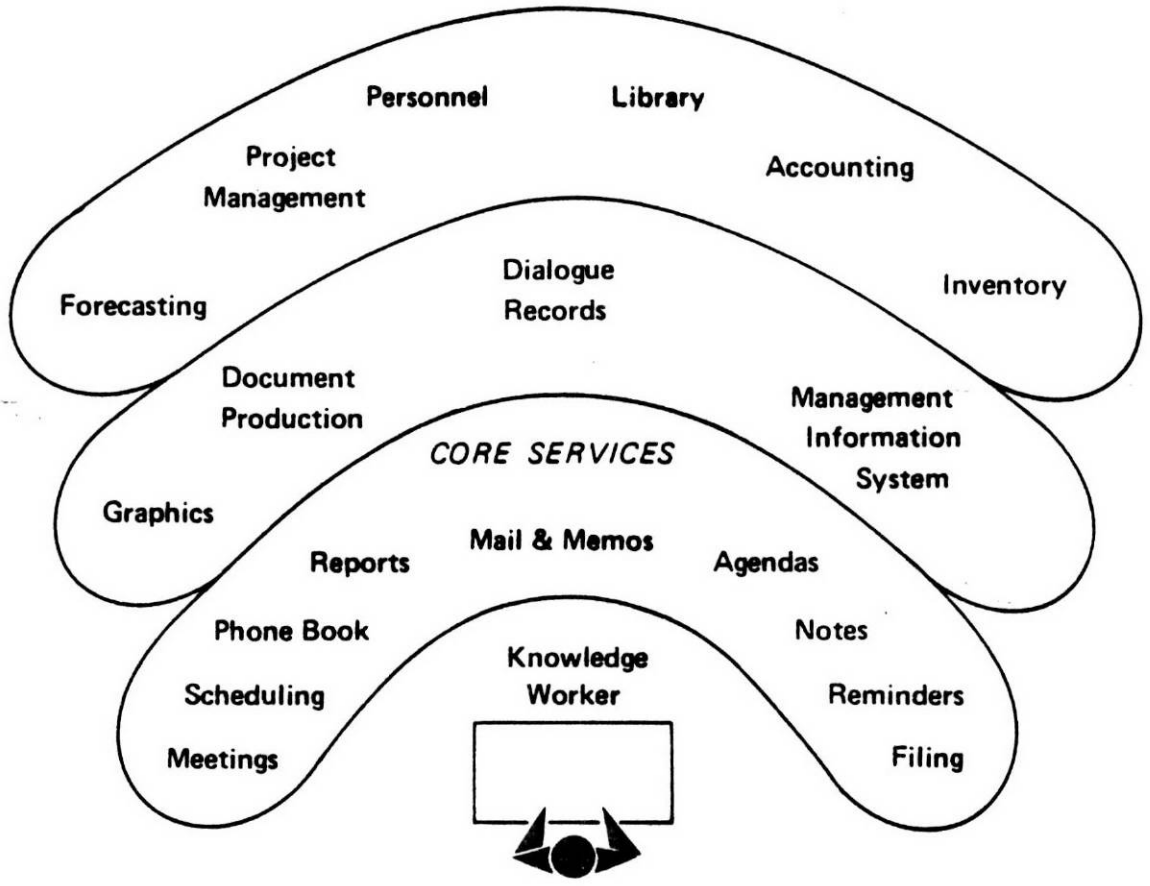


FIGURE A-1 INTEGRATED OFFICE-OF-THE-FUTURE SERVICES

specialized, less frequently accessed services are in the lobe that is most distant from him. The core services shown in the figure are most important in the effective performance of everyday activities. Core services are necessary for the individual and organization to function on a daily basis.

The support of core services may have the greatest potential payoff. Most of the daily activity is spent in providing these services. For example, communicating, using a core service, comprises almost 90% of a manager's time (Mintzberg, 1973). In addition, when core services are provided, accessing less used services is a straightforward extension of the same skills, techniques, and technology.

The office-of-the-future will provide all the services that each individual needs in one integrated system with one interface. The system will be designed to permit easy evolution to incorporate new technology and additional services. The integration into one system of present and future services is a key source of beneficial impact upon productivity.

Leverage of Impact Area Based on Labor Costs

The primary and most important question about the changes the office-of-the-future will bring is what are the benefits? Benefits may take many forms, for example, the reduction of menial tasks and the improvement of the quality of work. But the innovations required to bring about the changes require an investment, and thus there must be a return on the investment. Consequently, the benefits of greatest concern to institutions are cost savings, for example, a decrease in labor costs. Less tangible and much more difficult to relate to investment are qualitative changes. Thus, although improved quality is equally important, organizations are most concerned about quantitative benefits.

If the foregoing frame of reference is accepted, the second most important question is: where can the greatest quantitative benefits be realized? In other words, what area of change will have the greatest leverage. Current investment and technological development indicate that the answer is "word processing." There are countless implementations of hundreds of systems. Studies show that the primary activity in a word processing system, typing, can be improved by orders of magnitude, ranging from 200 to 500% in some cases (cf. O'Neal, 1976). Another major activity, author composition, can be improved 30 - 50% (according to IBM) through the use of machine dictation. These impressive results are well documented in the word processing community and well known by vendor sales people.

The productivity increases in word processing implementations are translated from increased number of lines or words per unit time (per hour or per day) into the amount of labor saved by the increased output. The point of leverage in this case is the typist's labor and the function is hardcopy production, either for general documentation or for correspondence. The reduction in the cost of labor is the return on investment.

However, if we look at the overall distribution of labor costs for U.S. Business (Figure A-2), we see that the costs of clerical support in general is about 34% of total labor. The secretary typists costs amount to only 6% of total labor costs. Thus, on the basis of the proportion of costs, the leverage of word processing is quite small. A quantitative savings in the secretarial-typists area will have a relatively low effect on overall operating expenses. This does not belittle the value of present approaches to word processing, but can serve to focus the investments on areas of greater potential return (Harkness, 1977).

Non-clerical labor costs are 66% of the total indicating that this area should be examined for potential quantitative benefits. The breakdown between management and professionals is interesting. However, we need to understand the activities of this labor sector and how they will change as a result of innovations. Typing is the activity most supported in the secretary typist area. What activities can be similarly supported in the other labor sectors? The clerical sector is supported by data processing which includes accounting, inventory, and other computer based functions. Since the non-clerical sector does not seem to be supported and yet comprises 66% of the costs, non-clerical activities appear to be most important to examine.

Leverage Based Upon Daily Activities

The distribution of daily activities for secretary-typists supports the notion that leverage for office automation is not in typing. Figure A-2 shows the average work distribution for secretary-typists. This survey data indicates that typing is approximately 20% of the on-the-job hours (Purchase, 1976). Thus, typing is approximately 1.2% of the labor costs of businesses (20% of the secretary-typists' labor which is 6% of total white collar labor costs). Although the percentage is extremely small, the dollar amount is 4.4 billion for all U.S. business in 1974. By comparison, non-clerical labor costs total $249 billion.

Non-clerical activities are divided into essentially two categories, managers and professionals. Managers' activities have been studied by several persons (as summarized by Panko, 1976) which all seem in general agreement. One of the most interesting and thorough of these studies, Mintzberg (1973), is the source of the managerial work distribution shown in Figure A-3. Oral communication (meetings and phone calls) amounts to 75% of what a manager does with his time. Desk work (22%) is described as mostly reading and writing (20% of the total), in other words, written communication. Based on Mintzberg's study of activities, 95% of managerial labor is spent in written and oral communication.

Professionals (non-managers in this case) tend to spend less time in communication. In a composite of three studies (over 5000 subjects) of scientists and technologists use of time (cited by Panko, 1976), 63% total was spent in communication which included 37% oral communication. I found similar results in studies of an Air Force research and development facility (Bair, 1974). A

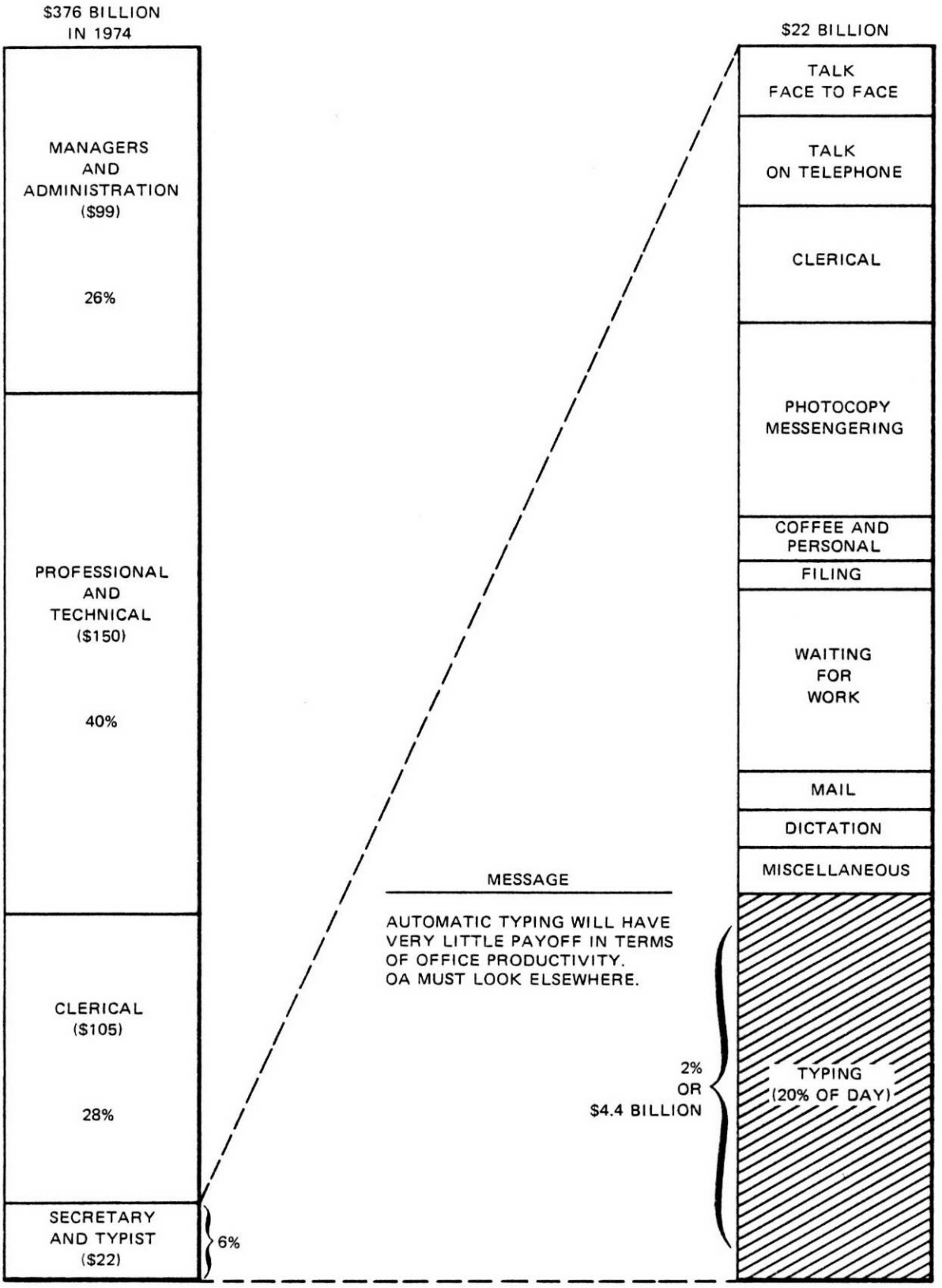
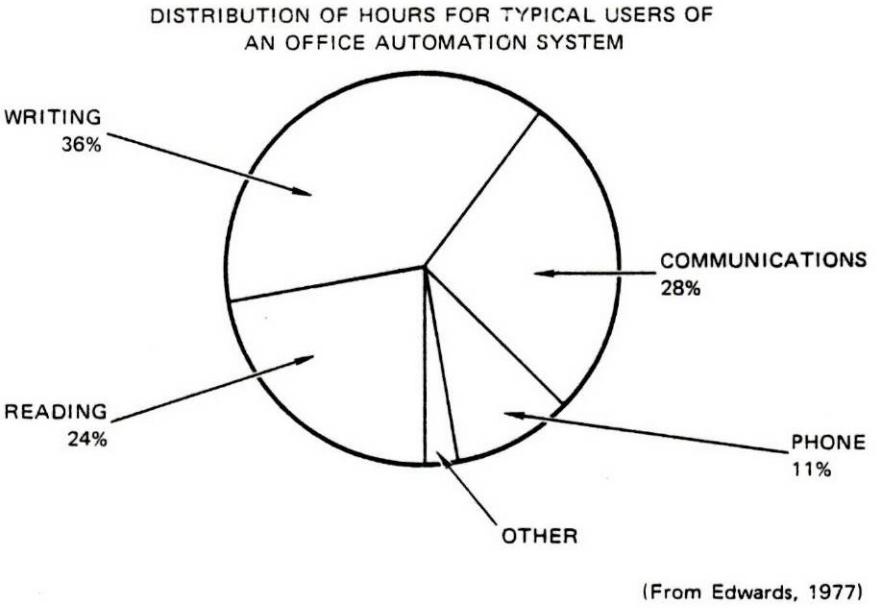


FIGURE A-2 LABOR COSTS FOR TYPING RELATIVE TO TOTAL OFFICE LABOR COSTS
(From Statistical Abstracts of the U.S., 1975)



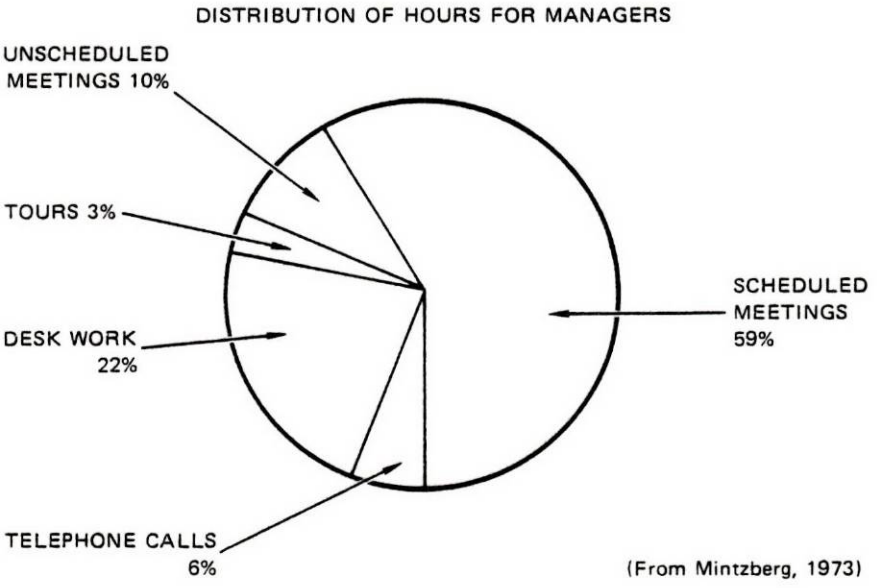


FIGURE A-3 DISTRIBUTION OF TIME AND ACTIVITIES BY MEDIA

survey of white collar workers in general--including secretarial--shows a similar distribution: 39% of daily activity is oral communication (Edwards, 1977).

The conclusion that labor costs are incurred largely by professionals and managers in the process of communicating raises additional questions. The most important is: what changes in the office-of-the-future could reduce this cost? The changes in activities based upon the new technology have been presented above. Since these changes are projected on the basis of years of experience with prototype systems, we can be reasonably certain that technology can be used to support the communication process (c.f. Bair, 1973). However, to address the foregoing question, we must consider whether technology could reduce the labor involved in communication.

Since non-clerical communication has such high economic leverage, a small percentage increase labor savings will have a high impact. This removes some of the pressure to show large magnitudes of savings to justify the investment in the office-of-the-future.

Some Benefits of Interpersonal Communication Technology

Benefits of the use of interpersonal communication computer technology in an office automation system have been observed through organizational studies (e.g., Bair and Conrath, 1974, and Carlisle, 1976) and several years of experience in the development and use of computer based message systems in the ARPA Network community. Other researchers have made similar observations (Turoff, 1977, Hiltz, 1977; Edwards, 1977). The following list of benefits is by no means comprehensive, and overlooks some of the problems that could possibly reduce the beneficial effects. However, some of the benefits could result in a reduction in labor.

TABLE A-1 LIST OF BENEFITS OF COMPUTER MAIL

1.	Permanent, searchable, stored record
2.	No simultaneous activity necessary
3.	No meeting schedule necessary
4.	Optimum time for composing, reading, and responding may be selected
5.	Physical collocation not necessary
6.	No interruptions
7.	One action for general information distribution
8.	Fast delivery at low costs
9.	Automatic distribution
10.	Automatic headers

This list of benefits is brief and does not discuss the broader implications for organizations and communities that use computer mail. There are other studies and reports that address the benefits and some of the problems of computer mail in greater depth, most notably Uhlig (1977) and Turoff (1977). Uhlig's experience is with a massive government organization and one of the largest computer mail system implementations to date, and bears out these benefits:

1. Permanent, searchable, stored record. A permanent, automatically stored copy of all communications is available for retrieval by keyword, sender, and other characteristics, and for subsequent use in other compositions.

2. No simultaneous activity necessary. The sender and receiver(s) do not need to interact simultaneously. Messages may be originated, sent, received, and read at different times (asynchronously), an important benefit when the geographical separation involves considerable differences in time zones (e.g., from London to San Francisco). Because messages may be sent or received at any time, portable terminals can be used at home, on trips, and during non-business hours.

3. No meeting schedule necessary. Topics can be introduced and be responded to by all discussants at their convenience, and over an extended period of time.

4. Optimum time for composing, reading, and responding may be selected. Communications may be composed, read, and responded to when the communicators are best able to respond, having the needed resources and disposition. Although formality is less than in a letter or memo, the printed text is available for careful composition and studied response.

5. Physical collocation not necessary. Physical location is not important in any communication, permitting interactions in situations in which telephoning is too costly and the mail service is too slow and/or unreliable.

6. No interruptions. Meetings and conversations are not subject to interruption that would hinder their progress, or cause ill feelings. Message preparation or reading can be handled out of the context of social pressures.

7. One action for general information distribution. Simple communications, such as an announcement, a request for information, or acknowledgments, require only one action. The problems of making contact by phone, such as busy signals, out-of-the-office, or preoccupied, are alleviated because the recipient need not be available. It is not necessary to leave messages that may be lost or misinterpreted, or to reschedule contact.

8. Fast delivery at low costs. The delivery time for written communications is reduced from days to seconds. The need for phone contact, with its inherent

obstacles and dependence upon synchronous contact, is minimized, and the utility of the written medium is increased greatly.

9. Automatic distribution. Distribution of communications is automatic through specification of recipients' names or through recipient lists. This may serve as a broadcast capability, which reaches all users with more assurance than the chance viewing of a bulletin board, and with less labor than the hand distribution of printed memos (extremely difficult for geographically separated recipients). Electronic distribution permits simultaneous delivery of time-critical messages, avoiding political repercussions if there are unequal delays for different recipients. Copies are more easily sent to potentially concerned parties, increasing the potential for keeping all persons informed.

10. Automatic headers. Message systems automatically maintain data about messages that also can be used for retrieval, and for automatic packaging of responses. Date and time sent, distribution lists, length, title, number, and so on, are employed automatically by the program when responding to a particular message. Retrieval can be on the basis of any of these fields, including titles that form headers that permit browsing without examining the entire contents of the messages.

Interpreting the value of these benefits is difficult for many reasons. Basically, it is because it is difficult to translate qualitative (soft) benefits in quantitative (hard) values. Some valuable work by Johansen, Vallee, and Palmer (1976) has shown an impact of computer message systems upon productivity. However, their work concentrates more on the group communication process, and we also need to show impact upon labor costs. There are numerous additional points regarding qualitative benefits, but what about direct, quantitative gains?

Projection of Labor Savings from Communication Benefits

There are four kinds of communication benefits that I think can be translated into labor savings: (1) reduced shadow functions, (2) optimized message cueing, (3) automation functions, and (4) reduced media transformations. Many of the benefits listed above fall into these categories, each of which means reduced labor costs. It is assumed that well known mitigating factors, such as training costs, technology transfer, and system reliability, have been normalized, and the office-of-the-future technology is operating successfully.

"Shadow functions" are the unforeseen, unpredictable, time consuming activities that are associated with accomplishing any task but do not contribute to productivity (Holtzman, 1976). They follow workers through all daily activities but usually are not noticed. For example, when making a phone call, there are several potential shadow functions: a misdialed number, a busy signal, the recipient temporarily out of the office, the recipient

gone for some period of time, a bad connection, and so on. Meetings are another example, including delays in arrival (traffic jams, not finding the meeting location), other delays in starting (one member of the group being late), and so on. Many of the functions are due to timing--and real-time communication requires timing.

Personal observations indicate roughly that 15 - 25% of time spent in communication activities is due to shadow functions. Variations are caused by different activity mixes, for example, shadow functions tend to be greatest for phone communication. One scenario is as follows: a call is placed (30 sec.), connection is made but it is the wrong number (+15 sec.), redial (+30 sec.), an operator answers (+10 sec.), the call is connected to the wrong office (+30 sec.), and switched to the correct office (+20 sec.), a secretary answers and requests identification (+15 sec.), the recipient gets to the phone (+15 sec.), conversation (360 sec.). In this typical scenario, it takes 165 seconds to make the connection for a 6 minute call (the average length according to Mintzberg, 1973). In these timings, 110 seconds or 31% are shadow functions. Using Mintzberg's figure of 16 calls per day, we lose about 30 minutes on shadow functions in one activity of the non-clerical worker's day.

Message cueing is a second area of benefits that can translate into actual labor savings based upon optimum contact time and elimination of interruptions. Messages, usually through phone calls, that are not cued for attention at appropriate times, but are delivered regardless of current situations, cause shadow functions and result in interruptions of ongoing work. Interruptions cause wait and recycle times--whatever activity was ongoing at the time has to wait, and the recipient has to recycle into the original activity before continuing. Without going into a detailed scenario, estimated wait times average 3 minutes, and recycle times, 5 minutes. Interruptions are so common in non-clerical work, that it tends to be "interrupt driven" (which may have a strong negative impact on the quality of work). Using the quantity of unscheduled meetings and phone communication as indicators of interruptions per day, there are at least 60 minutes of wait and recycle time per day.

The third area of benefits is the automation of several steps in the communication process. The steps include addressing, labeling, dating, formating, distributing, storing, signing, and others. Each of these can be done completely automatically by the computer in the future office. In the present office, it is very difficult to isolate the exact time required to accomplish each step. However, the typical letter (250 words) requires about 5 minutes to specify to a clerical worker the needed information (including review and signature time). The numbers of letters prepared varies greatly among non-clerical workers, however, we estimate that there are an average of two per day (10 minutes). In addition, there appears to be a large savings potential from automation for clerical costs where it takes 53 minutes to manually prepare the typical letter (Konkel, 1976).

Media transformations--changing the medium of the message--are necessary between speaking and writing, handwriting and typewriting, dictation and typewriting, phone and written, computer and hardcopy, and local copy and mailed copy, and so on. The reduced number of media transformations is a benefit that depends upon the extent of office automation. The more media that are automated within a single system, the fewer the transformations between different media. The primary support cost in communication activity is from preparation, packaging, and mailing written correspondence. However, there also is the cost of producing any record of oral communication--the translation of the oral into the written medium. Notes from meetings and phone calls usually require additional time to translate into storable, permanent form. This averages approximately a minute per phone call (added to the time indicated above). The average notation time for phone calls is about 16 minutes per day.

Messages are typically stored as paper media organized in the omnipresent file cabinet. In an automated office, the online storage would eliminate this "media management" task which is often done by non-clerical personnel. The time to store and retrieve messages is difficult to quantify. If secretaries accomplish this task, non-clerical time is required to specify which message (letter or note). In either case, the non-clerical time is about 2 minutes per message. With a minimum of 10 messages per day, 20 minutes of non-clerical time is consumed.

Media transformations generally involve clerical time more than non-clerical. "Word processing" systems are mostly to support secretarial labor. The savings from eliminating clerical support for written communications is important in the office-of-the-future. This is not easy to envision, but our experience has shown that managers and professionals prefer to process their own mail using some prototype systems because it is faster and easier than transferring the needed information to a secretary. Another important benefit, but one not easily quantifiable, is eliminating a tedious, repetitive task by enabling the person (message author) who needs the action done to do it himself.

In these four areas, conventional activities required to communicate in an average day amount to approximately 2 hours. According to Konkel (1976), the average hourly wage for non-clerical personnel is $9.64. Typical overhead and other salary related costs triple that figure to approximately $30 per hour. On a day-to-day basis, this brief analysis shows a potential cost saving of approximately $60 per day for each non-clerical employee or $15,000 per year. Carrying this projection to the extreme, two hours per day is one quarter of management and professional/technical labor which is $62.25 billion based on total U.S. annual labor costs in these areas (Figure A-2).

This very brief and superficial analysis based on work measurement and other studies is merely indicative of labor savings and is summarized in Figure A-4. There are profound implications that my studies and others (Hiltz, 1977) are just beginning to reveal--changes in the whole fabric of organizations and the sociology of communication. To date, I have yet to uncover

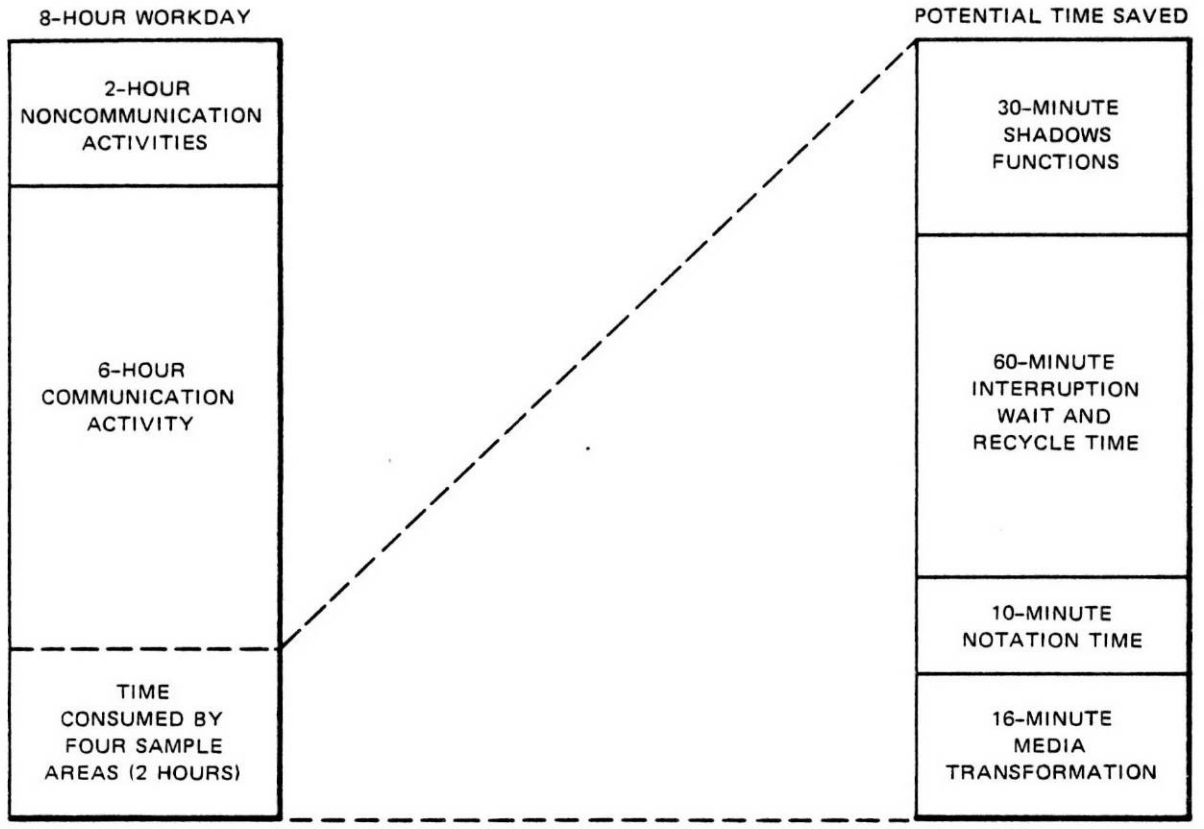


FIGURE A-4 NONCLERICAL LABOR BENEFITS FROM COMPUTER MAIL

negative consequences of computer mediated communication that would be cause for limiting usage. We do know that intelligent application couched in new discipline is warranted. As a minimum, the area deserves intense study.

Costs of Communication Technology

An assessment of the costs of the communications technology that makes the savings possible is necessary in order for projected labor savings to be meaningful. Two assessments of costs represent the range from high to low (Table A-2). The first is based upon analysis by Panko (1975) of six systems with an average message size of around 50 words. The second is from an updated projection by Baron (1974) of costs in a network environment.

TABLE A-2 COST ASSESSMENT FOR SENDING AVERAGE MESSAGE

	Panko	Baron (updated)
Preparation	$ .20	$1.30
Transmission	.30	.20
Terminal	.80	1.00
Total	$1.30	$2.50

Preparation, including composition and addressing, takes an average of 10 minutes. Transmission is instant and the terminal is intelligent. The cost of receiving a message is estimated to be approximately $1.00 (Baron). These figures may be optimistic due to the experimental nature of the technology. However, Hewlett-Packard's "COMSYS" and Tymnet's "OnTym" are production message systems where it costs less than a dollar per message, not including labor.

To generate a figure that can be used to compare service costs with cost savings, we can assume that during an hour of communication, the user will send and receive an average of at least four messages. This is a realistic average based upon experiences at SRI and a government study of computer augmented interpersonal communication (Bair and Conrath, 1977). Using a compromise figure of $2.00 per message sent, and $1.00 per message received, an hour's worth of messages would cost $6.00.

The projected labor savings above showed a 33% improvement in communication labor costs based on 6 hours of communication per day (75% of non-clerical time is communication activity), and a potential 2 hours per day saved. Using the hourly rate of $6, the daily costs would be $36. We showed a minimum cost saving of $60 per day (2 hours of non-clerical labor). The resultant overall savings from the simple analysis is $14 per non-clerical worker per day.

The conclusion that is warranted from this analysis is that computer based communication technology is cost effective. However, we do not have an order of magnitude saving based upon the conservative figures used here. $14 per day can be "lost in the noise." A cost-benefit analysis would probably show an improved return on the investment due to depreciation and amortization of technology costs--in five years the equipment costs could be paid off. But cost-effectiveness is only a small part of the picture. We have not quantified the many qualitative benefits discussed above which might be much more valuable than the four areas of labor savings presented here. In any case, we do not have to justify the qualitative benefits on a direct cost basis.

Measuring Changes in Interpersonal Communication

The foregoing discussion is about the "average" communication in an organization, but how can the actual communication activities be determined? The process of communication in organizations may be described, analyzed, and quantified using the communication audit methodology. The recommended version of this methodology has been developed under the auspices of the International Communication Association (ICA) and described by Goldhaber (1976). It includes four instruments: a survey questionnaire, interviews, critical incidents, and network analysis. The number of messages per unit time and other quantitative aspects of interpersonal communication that have been referred to above may be measured using network analysis. Network analysis provides the most quantitative data, whereas the other instruments help show cause and effect relationships. There are two well developed instrument-analysis packages for network analysis--one by Richards (1975) captures data through an interview like method which is reduced by the "Negopy" computer program. The second, the "communication tally" technique, was developed by Conrath (1972).

Conrath pioneered the use of network analysis for measuring the impact of computer communication. The use of the tally to measure the impact of a comprehensive computer communications system on an organization was accomplished by Bair and Conrath and reported to ICCC in 1974. A field experiment design permits comparison of organizational units using the technology and units not using the technology.

The tally captures the following data: the sender and receiver of messages, the length of all messages, and the modality. The mode refers to phone, written, and the different computer subsystems. The length of messages is translated into a standard unit for all modes. The data is analyzed using a standard computer program to develop networks which show the patterns of communication, clusters of communicants, and relationships both intra- and inter-organizationally.

The communication audit method for measuring the more labor related aspects of communication would include a structured observation or "activity sampling" instrument. This would capture the labor time spent in communication activities, and provides data such as that described earlier in this paper.

Mintzberg's descriptions of managerial work are based on this kind of instrument, and Carlisle (1976) has applied it to computer mail usage. Extreme care must be taken to ensure that the sampling periods are representative, which is much more difficult in the non-industrial settings of concern here.

The use of an early version of a communication audit method was done for the Air Force by Bair (1974). Using specially developed audit instruments supplemented by a controlled attitude test, I was able to show the changes in an organizational unit of about 20 persons over a three-year period. By comparing measurements with a like organizational unit that was not using computer technology for their work, causes of the measured changes were isolated. Much of the impact was due to specific aspects of the technology and the method of introducing it into the organization. However, in addition to the kinds of benefits described in this paper, there was one basic conclusion: interpersonal communication technology did improve the effectiveness of its users.

Limitations and Implications of Communication Technology

Having described the benefits, costs, and measurement of interpersonal communication by computer, several questions of major significance remain. This discussion is intended merely to acknowledge some of the problems in answering these questions--the issues raised will need sizable effort for some time to formulate answers.

The first question is, what about typing as an impediment to the use of computer message systems within the foreseeable state-of-the-art? Typing is indeed required by all users, although an intermediary between the terminal and the user is workable. In numerous experiences, systems have not been used because typing was demeaning and not a developed skill, as one might predict. But in as many cases, senior executives, administrators, and other "non-typists" have acquired the necessary typing skill with little bother in the process of using computer terminals. Also surprising is Chanpanis's conclusion (1977) that, "typing skill per se appears to be of little importance in the kind of interactive communication tasks tested in this experiment" (this so surprised the experimenters that additional computer communication experiments were run to validate the results!).

The reduction in labor costs through any enhancement of working methods is vulnerable to Parkinson's Law--will not the available amount of work expand to fill the available time? It will in some situations, but gains have been observed over long enough periods to have recorded this phenomenon if it were significant. Personal and organizational discipline coupled with all the other attributes of a healthy organization can minimize this inevitability.

Is communication through the new computer medium as effective as traditional modes of interaction? Any serious examination of this question will involve psychological and sociological investigations. Indications so far are that communicative

behavior will adapt to the computer medium as it did to the phone. Chapanis's studies (1977) found no difference between phone and face-to-face problem solving interaction, but these modes were twice as effective as the mediated communication in synchronous typewriter interaction. Operational use of a computer mail system at Bell Canada Business Planning in Montreal resulted in much more efficient administrative communication, but little support of problem-solving. Uhlig (1977) notes the miscommunications and other incidents resulting from the lack of non-verbal cues in interaction and the terseness of messages. Similar incidents occur throughout the ARPA Network community, but would oral communication have been different?

Perhaps the lack of understanding of media effect is resultant from the minimal level of involvement and attention to the issues by the behavioral and social sciences. Hiltz (1977) has made pertinent observations from a sociological perspective, and Edwards, Carlisle, Johansen and Vallee, Conrath, Bair and others have made contributions. But the most perceptive discussion is by Gerald Miller in a forthcoming book edited by Mead. Asking whether humanity and technology can peacefully coexist, Miller responds conditionally: if a harmonious communicative balance between human and mediated systems can be maintained. Indeed, I wonder if a certain level of face-to-face communication is fundamentally necessary to provide for human needs. Certainly more involvement and research is requisite to illuminating the neglected human factor in interpersonal communication technology.

Conclusion

This paper proposes a direct labor savings from the use of computer based mail systems. The logic begins at the global level of white-collar labor expenditures for the U.S.A. An examination of the proportion of time spent in different activities for each kind of white-collar labor shows that the greatest leverage for economic return is not in secretarial typing, but in non-clerical interpersonal communication. Then a detailed breakdown of communication activities into constituent steps is used to project the time consumed by unnecessary or inefficient steps ("shadow functions"). This time is costed on the basis of average salary and compared to the cost of the computer mail technology--computer mail appears to be cost-effective. The percentage of time that could be saved from the benefits of computer mail is the basis of an inductive leap back to U.S. white-collar labor costs.

Whether or not the U.S. or other nations can save billions of dollars in white-collar communication labor costs is speculative out of proportion to the rest of the paper, and is presented because it may be interesting. There are many mitigating factors affecting white-collar productivity, both pro and con. One of the most important is the effect of labor reduction on unemployment. If increased productivity means unemployment, we have a "catch-22" effect. On the other hand, low productivity in the white collar workers seems to be contributing to a general weakening of the U.S. economy, particularly when compared to more industrious nations such as Japan. If increased productivity can

support a shorter work week, what will happen to inflation? Discussion of the social and economic factors of productivity changes is almost overwhelming, seemingly outstripping our abilities to deal with enormously complex systems. However, the improvement and evolution of technology to support information processing and communication is inevitable, and the proposition that it is cost effective is encouraging.

REFERENCES

Bair, James H., "Evaluation and Analysis of an Augmented Knowledge Workshop." Final Report for Phase I, RADC-TR-74-79, Rome Air Development Center, April 1974.

Bair, James H., "Experiences with an Augmented Human Intellect System: Computer Mediated Communication." Proceedings of the Society for Information Display Journal, Vol. 14/2 Second Quarter, 1973, pp. 42-51.

Bair, James H. and David W. Conrath, "The Computer As an Interpersonal Communication Device: A Study of Augmentation Technology and Its Apparent Impact on Organizational Communication," Proceedings of the Second International Conference on Computer Communications, Stockholm, Sweden, August 1974.

Baran, Paul, "ARPANET Management Study: New Application Areas, Second Quarterly Technical Report," ARPA Order 2317, Cabledata Associates, Inc. Report R-160, August 5, 1974.

Carlisle, James H., "Evaluating the Impact of Office Automation on Top Management Communication," Proceedings of the National Computer Conference, New York, 1976.

Chapanis, Alphonse, et al., "Studies in the Interactive Communication: II. The Effects of Four Communication Modes on the Linguistic Performance of Teams during Cooperative Problem Solving," Human Factors, 1977, 19(2), pp. 101-126.

Conrath, David W., "Measuring the Computer's Impact on Organizational Structure." Stanley Winkler (Ed.), Computer Communication, First International Conference on Computer Communication, Washington, D.C., October 24-26, 1972, pp. 68-73.

Edwards, G., "An Analysis of Usage and Related Perceptions of NLS--a computer based text processing system," H. Q. Business Development, Bell Canada, Montreal, Quebec, October 1977.

Engelbart, D. C., "Augmenting Human Intellect: A Conceptual Framework." Stanford Research Institute, Menlo Park, Calif., Summary Report, October 1962.

Goldhaber, Gerald M., "The ICA Communication Audit: Rationale and Development." Paper presented at the Academy of Management Convention, Kansas City, August 1976.

Harkness, Richard C., Satellite Business Systems Corp., McLean, Virginia, Personal Communications, 1977.

Hiltz, Roxanne Starr. "Computer Conferencing: Assessing the Social Impact of a New Communications Medium," Technological Forecasting and Social Change, 10, 1977, pp. 225-238.

Holtzman, David L. and V. Rosenberg., "Understanding Shadow Functions, the Key to System Design and Evaluation." Paper presented to the Xerox Workshop on Evaluating the Impact of Office Automation, Xerox Palo Alto Research Center, Palo Alto, Calif., May 13-14, 1976.

Johansen, Robert, Jacques Vallee, and Michael Palmer, "Computer Conferencing: Measurable Effects on Working Patterns." Paper prepared for the National Telecommunications Conference of the Institute for Electrical and Electronic Engineers. Menlo Park, Calif.: Institute for the Future, November 29, 1976.

Konkel, Gilbert J. and Phillis J. Peck, "Traditional Secretarial Cost Compared to Word Processing." The Office, February, 1976, pp. 67-68.

Licklider, J.C.R., "Interactive Information Processing," in: Julius T. Tow (Ed.), Computer and Information Sciences-II. N.Y.: Academic Press, Proceedings of Second Symposium on Computer and Information Science, 1966.

Miller, Gerald R., "Communication in the Third 100 Years: Can Humanity and Technology Coexist Peacefully?" in C. David Mead (Ed.), Life in America in 2076. East Lansing, Michigan: Michigan University Press, IN PRESS.

Mintzberg, Henry, "The Nature of Managerial Work." N.Y.: Harper and Row, 1973.

O'Neal, Jeffrey D., "We Increased Typing Productivity 340%," The Office, February, 1976, pp. 95-97.

Panko, Raymond R., "The Outlook for Computer Mail," Journal of Telecommunications Policy, June 1977.

Panko, Raymond R., Presentation to the Seminar on the Augmented Knowledge Workshop, Stanford Research Institute, November 1976.

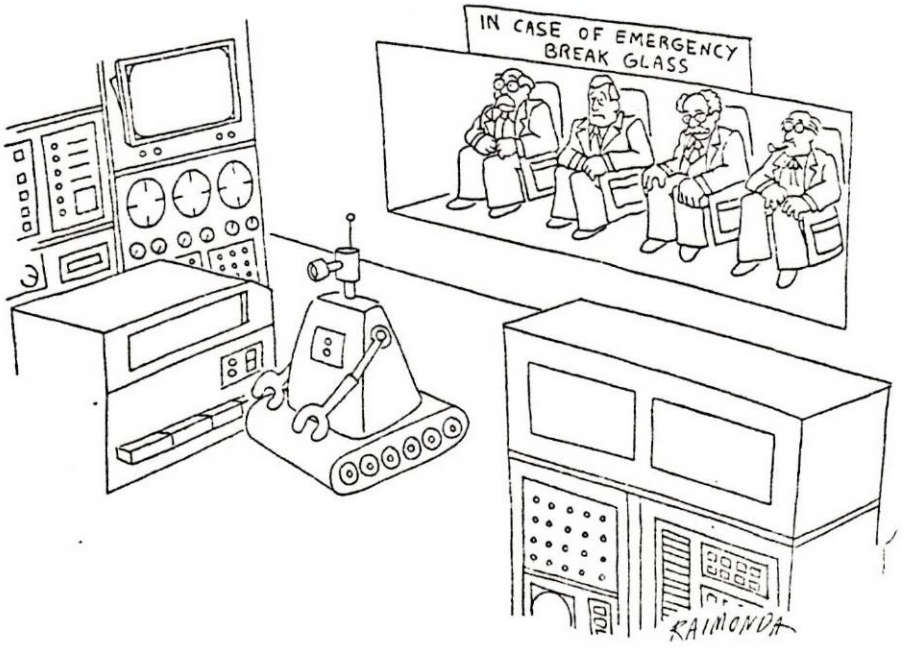
Purchase, Alan, "Office of the Future," SRI International Business Intelligence Program, Guideline No. 1001, 1976.

Richards, William W., A Manual for Network Analysis: Using the Negopy Network Analysis Program. Stanford, Calif.: Institute for Communication Research, Stanford University, Calif., 1975.

Statistical Abstract of the U.S., U.S. Depart. of Commerce, Bureau of the Census, 1975.

Turoff, Murray and Roxanne Hiltz, "Meeting Through Your Computer," IEEE Spectrum, pp. 58-64, May 1977.

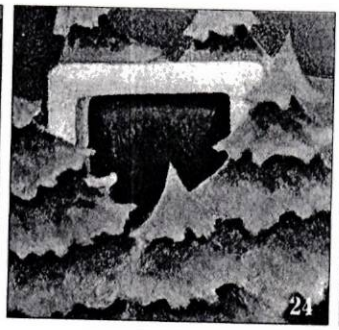
Uhlig, Ronald P., "Human Factors in Computer Message Systems," Datamation, pp. 121-126, May 1977.



JANUARY 1987

HIGH TECHNOLOGY

VOL. 7, NO. 1



FEATURES

DEPARTMENTS

THE ELECTRONIC MAILBOX: AS CLOSE AS YOUR PC

Corporate and public message networks continue to grow as services become easier to use

When Stan Prochaska, a public affairs supervisor at the U.S. Department of Agriculture in Washington, D.C., sits down at his desk in the morning, he ignores his in-box. "First thing I do," he says, "is check my electronic mailbox—often there are as many as 20 messages waiting for me—and then I answer my mail from my terminal."

When Prochaska logs on to an electronic mail service provided by Dialcom (Silver Spring, Md.) from his desktop personal computer, he scans the contents of his mailbox, deciding which items to read, which to hold for later, and which to discard, all with a few keystrokes. A request for information on pesticides from an agricultural extension office in Orlando, Fla., might be answered by a brief typed reply, together with electronic copies of press releases and news items located by a quick search of the USDA's database. A message that includes information on crop yields from the USDA's office in Lincoln, Neb., can be copied onto a floppy disk for inclusion in a press release, which will later be disseminated by the electronic mail system.

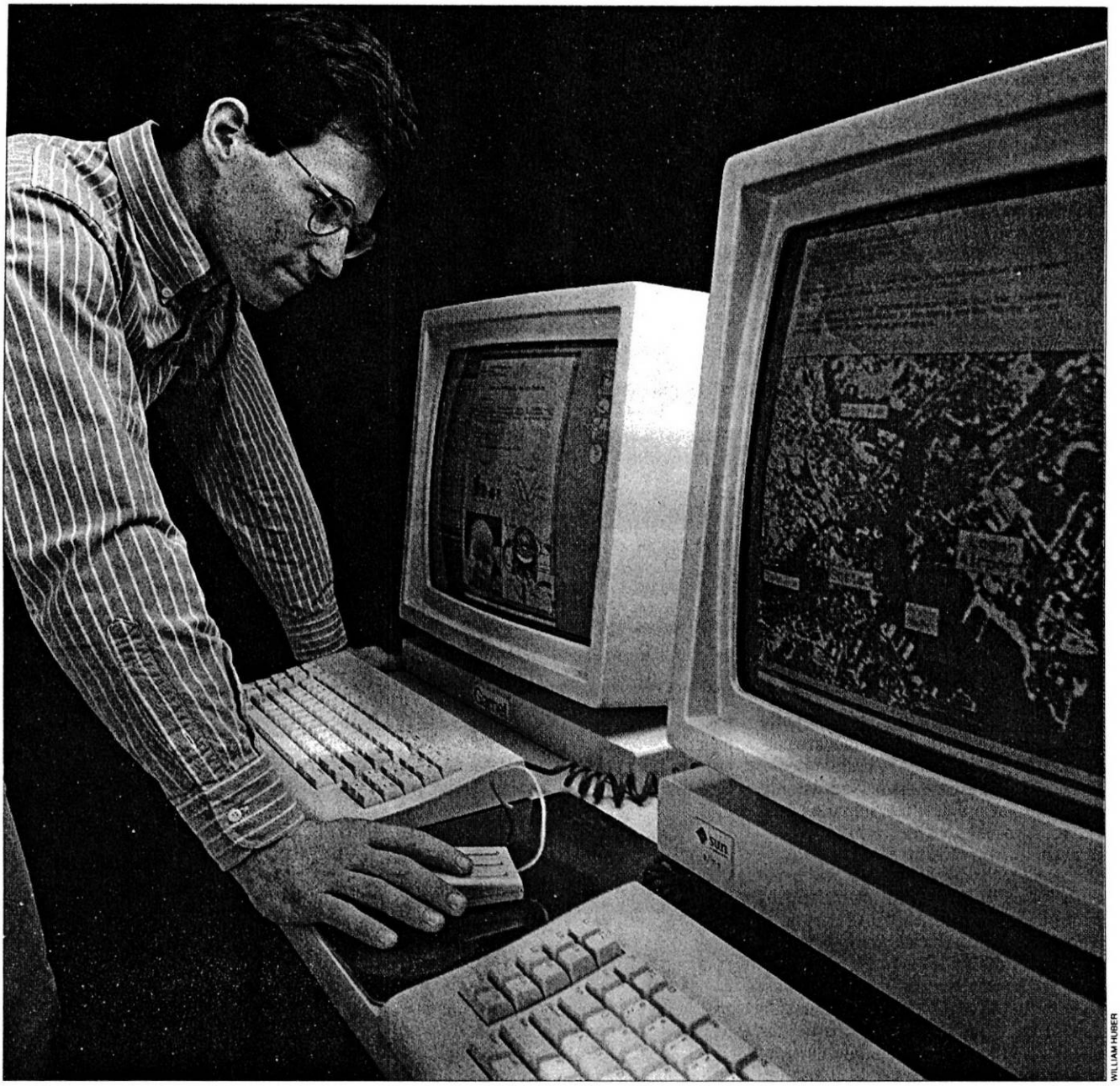
Prochaska logs on several times a day. "People who want to reach me know they can do it faster and more reliably this way than by telephone," he says. "When I travel, I take a portable computer and dial in to my electronic mailbox from the motel or even an airport lounge."

Despite a growing number of confirmed users of electronic mail—computer-based messaging systems that let users send documents and data to other users, for retrieval at will, over corporate or even worldwide networks—the spread of such services has been slower than early proponents predicted. The first public systems were hard to use and thus had few subscribers. But recently developed software for terminals and personal computers is making the writing, sending, and receiving of electronic correspondence easier, faster, and cheaper than regular postal mail. And as private electronic mail is becoming the backbone of networked office automation systems, public electronic mail systems are scrambling to link up with major corporate networks and even with each other. As a result, electronic mail use appears to be approaching the critical mass that could make it the medium of choice for business communications in the near future.

The concept of electronic mail has been around for more than a decade. Users of the Defense Department's ARPAnet have had a nationwide mail network for several years based on computers at universities and laboratories. Users of large time-shared computer systems have long been able to send messages to other users, as have subscribers to remote computing services such as CompuServe (Columbus, Ohio) and The Source (McLean, Va.).

But the real growth has come only in recent years, as major corporations have adopted sophisticated office automation systems with electronic mail facilities. The computers that many businesses bought in the early 1980s largely for word processing have now evolved into versatile and extensive systems connecting

by Jeffrey Bairstow



BBN's Terry Crowley checks an experimental "multimedia" electronic mail system that combines text, graphics, photographic images, spreadsheets, and voice.

hundreds of users to computer networks that can span a single office or an entire company with worldwide offices and plants. Typically these systems integrate several applications, such as word processing, calendar scheduling, database management, financial spreadsheets, computer programming, and data processing. In addition, says Edward Thomas, manager of office automation systems for Data General (Westboro, Mass.), "we find that customers [for office automation systems] are demanding networking and electronic mail."

Thomas also points out that more business managers and executives are prepared to use a computer terminal or a PC themselves instead of delegating the work to a secretary. For such executives, reliable and rapid communication with other managers is of prime importance. Thus electronic mail is becoming well established in large and medium-size companies. Altogether, there are about 4 million in-house electronic mailboxes in the U.S., estimates Michael Cavanagh, executive director of the Electronic Mail Association (EMA), an industry trade group based in Washington, D.C.

Typical of the electronic mail facilities offered by office automation packages is the one included with Data General's Comprehensive Electronic Office (CEO). A user, after logging on to the computer system from a desktop terminal or a personal computer, is presented with a menu of options. Selecting electronic mail allows the user to compose messages that can be sent to other users on the same computer or on a network. Alternatively, documents previously prepared with the CEO word processor or spreadsheet program can be included in an electronic letter. Each mail user has an electronic inbox where messages accumulate for reading at a convenient time. A user can then read, answer, forward, file, print, or delete messages with a few keystrokes.

Some other office automation systems

with sophisticated electronic mail facilities are All-In-One from Digital Equipment Corp., PROFS (Professional Office System) and DISOSS (Distributed Office Support System) from IBM, and Wang Office from Wang Laboratories.

While in-house electronic mail has been spreading rapidly, the growth of public electronic mail services has been slower, according to EMA's Cavanagh. There are only about a million subscribers to the dozen or so public electronic mail services in the U.S., provided by MCI, Western Union, General Electric, Telenet (US Sprint), and others. Although messaging was provided earlier by some computer timesharing services, large public electronic mail services date from about 1982, when Western Union (Upper Saddle River, N.J.) first introduced its EasyLink service, followed by MCI (Washington, D.C.) with MCI Mail. Revenues from electronic mail services are estimated at $270 million in 1986, according to International Resource Development (Norwalk, Conn.), and are expected to grow to $1.2 billion in 1990.

Some 200 million pieces of public electronic mail were dispatched last year, says Eric Arnum, editor of *Electronic Mail and Micro Systems* (*EMMS*), an industry newsletter. By contrast, 140 billion pieces of mail are carried annually by the U.S. Postal Service—about 560 pieces per person, versus only about 200 per public electronic mail subscriber. A recent study by Venture Development Corp. (Framingham, Mass.) summed up the problem with one basic truth: "The primary reason people are not using electronic mail is because they can't communicate with the people they need to communicate with."

Until recently, purely electronic messages could be sent only to another subscriber on the same system. To ease this limitation, virtually all the public electronic mail systems are linked to telex networks, so a message can be sent directly to a telex subscriber (and vice versa, in most cases). If the addressee subscribes to neither electronic mail nor telex, messages can be printed out at one of the service provider's centers, usually by high-speed laser printer on a facsimile of the user's own letterhead, and sent by regular mail or courier to a conventional mail address. But in many cases, this method of delivery is no faster than regular mail or a package service, and may be even more expensive.

In recent months, however, there has been a flurry of announcements, both by private electronic mail vendors and by public service providers, of new interconnections among electronic mail systems. For example, Wang Laboratories (Lowell, Mass.) now offers a gateway, or specialized communications package, that allows electronic mail to be sent between users of its VS Office and users of the popular PROFS office automation package, which runs on large IBM computers. Messages can be sent without added complications or special routines, says Jacqueline Appel, office automation sales manager for Wang.

Digital Equipment Corp. (Merrimack, N.H.) offers a similar gateway, VAX Mailgate, that permits DEC VAX computers running the All-In-One office automation system to communicate with MCI Mail subscribers and services. MCI also provides independent software developers with technical support to encourage them to write links to MCI Mail for private electronic mail systems. One such firm, SoftSwitch (King of Prussia, Pa.), recently announced a link to MCI Mail for IBM mainframe computers running PROFS.

Western Union has gone a step farther, offering two types of links between its EasyLink electronic mail network and IBM's PROFS and DISOSS office automation systems. The software for these links resides on EasyLink computers, so IBM users need not install non-IBM software on their systems. One PROFS interface is a high-speed connection, for high-volume users, that performs as another (multiuser) node on an IBM PROFS network. The other interface, for lower-volume users, appears to the IBM system as a single remote terminal. In neither case are any modifications to the IBM systems required—a major selling point for data processing executives who wish to keep their systems "pure" IBM. AT&T Mail (Basking Ridge, N.J.) offers similar benefits to UNIX users. Any computer system running the UNIX operating system can connect to AT&T Mail via the electronic mail software routines normally provided as part of the operating system.

The most significant extension of electronic mail to date came with a recent internetwork link that allows the 75,000 MCI Mail subscribers to communicate with the 275,000 users of CompuServe's EasyPlex remote access services. MCI has also announced an overseas link to France's Missive service. And Telenet (Reston, Va.), the provider of the U.S.-based Telemail service, began a link to Telecom Canada's Envoy service to give Canadian electronic mail subscribers access to Telemail's worldwide network.

Despite all this progress, most of these links require proprietary software and are specially developed for each application. "The interconnection of electronic mail systems is not yet as straightforward as the interconnection of telephone systems," says Douglas Brackbill, senior marketing manager for MCI. "Most links do not conform to international standards."

But work on such standards is moving forward. The Consultative Committee on International Telephony and Telegraphy (CCITT), an international standards body, has had a study group working on electronic mail standards since 1981. The CCITT Message Handling Standards group has produced a set of interoperability standards known as X.400, based on the protocols of the Open Systems Interconnect (OSI) model (HIGH TECHNOLOGY, Sept. 1986, p. 30). The X.400 standards have been adopted by CCITT and endorsed by the U.S. National Bureau of Standards, as well as by the Corporation for Open Systems (COS), the U.S. computer industry group formed to encourage communications compatibility for computer systems. COS is expected to establish a test laboratory to verify that X.400-based products will operate as intended.

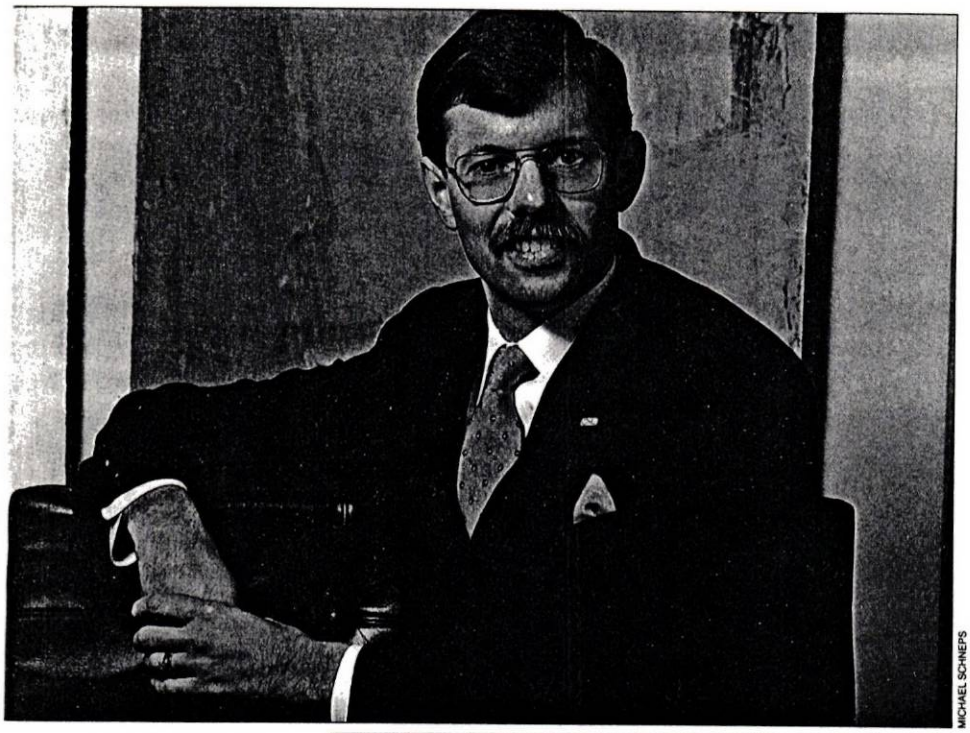
Pressure to adopt X.400 appears stronger in Europe, where electronic mail systems are often part of the national telephone companies (PTTs) and thus more closely regulated than in the United States. One international electronic mail service provider, Dialcom, a former ITT subsidiary now owned by British Telecom, has demonstrated an X.400-compatible electronic mail system, and other providers are expected to follow suit shortly. Dialcom has a strong incentive to adopt X.400, since much of its business is derived from licenses for its electronic mail services to PTTs in Europe and Asia. Ensuring compatibility between these PTT systems and private electronic mail systems is a priority for Dialcom, says John Morris, the company's president.

But in the U.S., where IBM holds sway over computer communications and network architecture, X.400 is faring less well. Computer links must be compatible with IBM's System Network Architecture

THE PUBLIC ELECTRONIC MAILBOX POPULATION

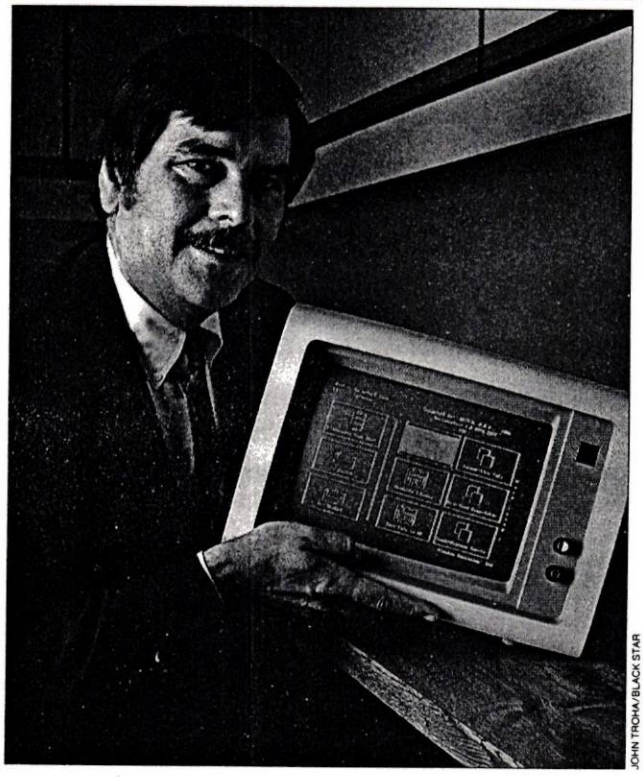
Company	Subscribers
CompuServe EasyPlex	275,000
WU EasyLink	130,000
Dialcom (British Telecom)	100,000
OnTyme (McDonnell Douglas)	85,000
MCI Mail	75,000
Telemail (US Sprint)	70,000
GE Quick-Comm	65,000
CompuServe InfoPlex	30,000
RCA Mail	25,000
AT&T Mail	20,000
Others	100,000
TOTAL	975,000

SOURCE: THE YANKEE GROUP



MICHAEL SCHNEPS

Macintosh-like PC software makes electronic mail easier for executives, claims GE's Norman McBurney.



JOHN TROHA/BLACK STAR

"Interconnection is the issue," says Walter Ulrich of Coopers and Lybrand, "and standards are forcing it."

(SNA), which is incompatible with the OSI model; and electronic mail must be compatible with IBM's Document Content and Interchange Architecture (DCA/DIA), which is incompatible with X.400.

Some computer industry experts believe that these factors could slow the growth of electronic mail in the U.S. "We do not expect to see a resolution [between IBM's architectures and X.400] very soon," says Joseph Forgione, group manager of communications for Data General. "In fact, the two standards will probably exist well into the 1990s." Thus Data General and other non-IBM manufacturers expect that they will have to provide both IBM and X.400 links for their computer systems. Independent software developers are already active in this area. One of the first X.400-based messaging software products for in-house use has been introduced by Sydney Development Corp. (Vancouver, B.C.), based on work done at the University of British Columbia. Sydney's software also has gateways to IBM's PROFS and the MS-DOS operating system, an important requirement for large companies with IBM mainframes and personal computers.

IBM will not comment on its plans for future products, but the company is a member of COS—an affiliation that may hint at some eventual accommodation with OSI and X.400. For example, "IBM could make parts of its proprietary SNA protocols public property under the OSI banner," says Arnum of *EMMS*. The opening of SNA would certainly speed the interconnection of computer systems and networks for non-IBM manufacturers.

CCITT is also developing standards for directories. With an electronic mail system on a single computer, a central directory of users is easy to set up and maintain. In a complex electronic mail network with many computers in several locations, accessing and updating a global directory becomes a massive task; so a more realistic solution is to have smaller local, or "distributed," directories. There are still no generally accepted standards either for directories or for methods of addressing to find subscribers. But a CCITT study group has been examining the problem for almost three years. "I expect to see standards [for distributed directories] within a year," says Richard Miller, president of Telematics International (Palo Alto, Cal.) and a former delegate to CCITT.

The adoption of standards for directories could raise some difficult privacy questions, says Walter Ulrich, a partner at Coopers and Lybrand (Houston). "A company might want to make its directory available to customers but not, say, to a firm of headhunters," notes Ulrich. "When there is widespread interconnection of electronic mail systems, there is potential for misuse of directory information." One solution might be for a company to publish its electronic mail address but not its internal directory. Thus an employee might give his mailbox address only to specific outsiders, who would keep their own personal directories on a PC, much as individuals now keep a personal telephone book.

Standardization may help to expand the universe of subscribers, but electronic mail users also need improved screen displays, or user interfaces. Early systems often followed a telex-like pattern and required the use of codes and abbreviations that had to be followed exactly. "Electronic mail is, for the most part, still difficult to use," says Jack Nilles, director of

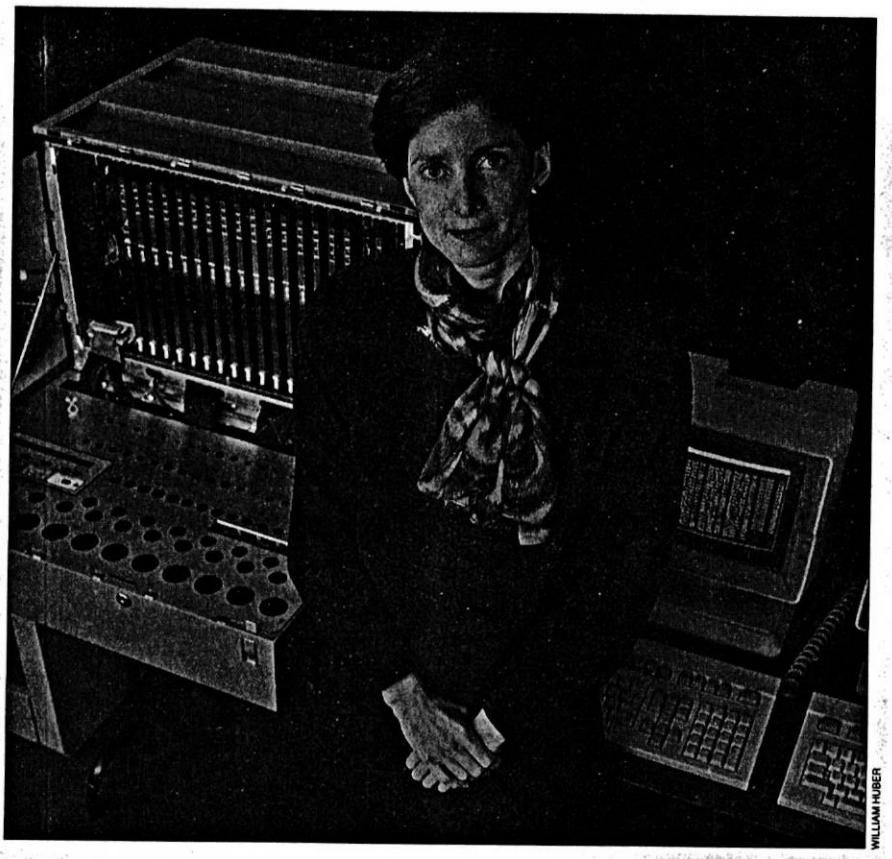
the information technology program at the University of Southern California. "It is more user-surly than user-friendly." But vendors are working on simpler interfaces, particularly for PC users.

Western Union, for example, has a software package called Instant Mail Manager, developed by Kensington Microware (New York). The package, which contains word processing, list management, file maintenance, and communications functions on a single floppy disk, allows a user of Western Union's EasyLink service to prepare mail off line with the word processor and send and receive mail automatically. The software package will call Easylink at a predetermined time, log on, transmit any prepared messages, scan the user's electronic mailbox, read and download any unread messages (to a floppy or hard disk), and log off. Electronic mail can then be read and rerouted at leisure. An extension to Instant Mail Manager, called Instant Forms Plus, lets EasyLink users design and transmit their own electronic business forms for applications such as order entry and invoicing.

Another popular software package, Transend PC from Transend Corp. (Portola Valley, Cal.), uses graphics to simulate a series of in- and out-boxes and files on the display screen. It also calls specified electronic mail services automatically, mailing letters and recording incoming mail on disk. The package has proved so easy to learn that it has been adopted and modified by several electronic mail services for use by their customers with PCs.

A more advanced alternative for executives and managers is GE's BusinessTalk system, a software package for the Apple Macintosh or the IBM PC and compatibles that combines electronic mail with database information retrieval and management and a private bulletin board system. According to Norman McBurney, manager of technical marketing at General Electric Information Services Co., BusinessTalk evolved from AppleTalk, a private system that the company developed for Apple to allow it to communicate with its dealers nationwide and around the world. In adapting this software to the IBM PC, GE has retained the Macintosh emphasis on graphic displays and easy interaction with the user.

McBurney describes BusinessTalk as a value-added service, one that provides more specific utility than basic electronic mail or remote computing service—and that presumably allows the provider to charge more for the service. Indeed, all the electronic mail services are anxious to pursue add-on applications. MCI Mail even has a "value-added reseller" program that lets independent companies provide applications and resell MCI's service.



WILLIAM HUBER

"Electronic mailboxes can talk," says Wang's Martha Danly, "but converting speech to text is tough."

Most of the public electronic mail systems are currently limited to transmitting text messages encoded in ASCII format—the American Standard Code for Information Interchange—which permits simple error checking and is relatively reliable for document transmission. This means that files in a more condensed binary format, such as computer programs or spreadsheet models, cannot be sent directly. GE's BusinessTalk system, however, has an error detection and correction protocol that can send binary files. Thus a user can develop a spreadsheet model or a Macintosh graphic and send the complete file to another user. The recipient can then work directly on the file and then return it to the sender for further modification. AT&T Mail has a similar binary file transfer facility.

In the future, electronic mail systems may be able to combine text, graphics, photographic images, spreadsheet models, and voice within a single message. At BBN Laboratories (Cambridge, Mass.), researchers have designed, and routinely use, such a multimedia system, called Diamond. Developed for the Defense Advanced Research Projects Agency, the system operates on a high-performance workstation, currently a model from Sun Microsystems (Mountain View, Cal.) that is capable of displaying and editing such multimedia documents. Terry Crowley, a BBN researcher working on the Diamond project, admits that the high cost of suitable workstations is an impediment to widespread use of multimedia systems, but he notes that workstation prices are now dropping (to as low as $10,000) and that the next generation of personal computers will be powerful enough to run such software.

Some aspects of multimedia mail are al-

SEALED WITH A SMILE

Among aficionados of electronic mail, a convention has arisen for indicating a joke: inserting "smileys," combinations of characters that, when turned 90°, look like tiny smiling faces. Some smileys we've seen:

:-) the standard smile

;-) a winking smile

B-) a smile with eyeglasses

8-) a smile with granny glasses

:@) a smile with a big nose

And then, of course, there is the opposite emotion :-(

—*Jeffrey Bairstow* B-)

A TRIP TO THE ELECTRONIC POST OFFICE

Because I work out of my home, I've been using an electronic mail service—MCI Mail—for the last couple of years in order to send text to (and receive text from) HIGH TECHNOLOGY's Boston office. In addition, it helps me to keep in touch with other writers and sources of information, and even to send letters to those unfortunate people who don't have an electronic mailbox. All my writing is done on a personal computer, so I compose drafts of letters—whether intended for electronic or hard-copy delivery—with a word processor and store them on disk. Not only is it much easier to edit with the word processor instead of on line to the electronic mail service, but I can transmit letters much faster directly from the disk and so save connection time (and money).

Generally, I check my mailbox once a day. My word processor (XY-Write) lets me call up my communications program (Perfect Link) directly. The latter has MCI Mail's phone number, my mail user name, and my confidential password stored in an automatic log-on sequence. The program dials MCI Mail through a modem, waits for an answer, and then logs me on without my intervention. MCI Mail responds with its own identification and a message telling me how many letters are in my mailbox. Usually there are four or five, but I've had as many as 30 on occasion.

At this point, most mail services offer a menu—scan the mailbox, read a letter, create a letter, and so on—but as an (ahem!) advanced user, I've opted for MCI's service that uses two-letter commands and dispenses with repetitious, boring, and time-consuming menus. To look at incoming mail, I type *SC* (for scan), which produces a numbered list of the letters in my in-box that shows the sender, the subject, the length (in characters), and the date and time of sending.

To look at a particular letter, I type *RE 1* (read letter 1). A header is displayed on the screen, giving the sender's name, MCI address, and affiliation, as well as the subject and the number of "carbon copies"—followed by the text, one screenful at a time. Short letters I often answer directly by using the command *AN* (answer), which automatically sets up a header with the recipient's name and my name (*To:*, *From:*, etc.), and then lets me type my reply directly. Longer letters or manuscripts from authors I store on disk with the *PR* (print) command, which scrolls a message across the screen without stopping, and the file capture routine in my communications software. I occasionally *FO*rward letters to other people for their attention, often with a short note of my own attached.

To send letters (other than replies), I use the *CR*eate command, which then asks the name of the recipient and checks the MCI Mail directory to see if that person is listed. If not, I have the option of sending a paper letter to a conventional mail address. MCI Mail has several paper mail options including four-hour or overnight delivery by courier service as well as delivery by USPS. In all cases, an electronic copy of my letter is sent to an MCI Mail center near the intended recipient and output by a laser printer using facsimiles of my letterhead and my signature.

I can type the text of a letter directly or I can send a prepared letter by calling up the file from a disk with my communications software. Letters are closed with a slash (/) on a new line, and MCI Mail then asks if I want to send the letter immediately. Usually I do just that, but I could also edit the draft further at this point using MCI's rather clumsy line-oriented editor. When a letter is posted, MCI also sends a copy to my electronic out-box, where it will stay for five days—a useful feature in case a letter gets garbled in transmission and needs to be re-sent.

When I've completed my mail, I *EX*it from MCI Mail, and my communications software returns me automatically to my word processor. A typical session with four or five letters to read and a similar number of brief responses might take 15–30 minutes. That's faster than going to my local post office and a lot more pleasant. But ironically, my bills don't come in electronic form—even MCI does its billing by snail mail.—***Jeffrey Bairstow***

ready starting to appear. Users of AT&T Mail can call the system with a push-button telephone and have their mail read to them by a speech synthesizer. A recently announced option with the Wang Office system offers the same feature. However, converting telephone speech messages into electronic text is "several years away," says Martha Danly, Wang's voice products planning manager.

As electronic mail spreads and the subscriber bases grow, the sheer quantity of computer-based messages could become overwhelming, says Thomas Malone, a professor at MIT's Sloan School of Management. Malone and his colleagues are working on a system called the Information Lens to help people filter, sort, and prioritize electronic messages. For example, a social filter might give high priority to messages from an immediate superior. Or an economic filter might give high priority to messages to which the sender had affixed the electronic equivalent of first-class postage.

Malone foresees intelligent electronic mail systems that might actually "read" the mail and take actions based on rules developed by individual subscribers. One example is the use of message templates for sending information about meetings. If the request to hold a meeting is sent in a standard electronic form, a system like the Information Lens could be set up to determine who should get the message, automatically check the recipient's calendar, update the calendar with the meeting date and time, advise the recipient of the calendar update, and notify the meeting organizer that the recipient has seen the message and can go to the meeting. Malone expects to add rules to the Information Lens so recipients of electronic mail can build their own filters for several types of messages.

Before multimedia and intelligent mail systems move out of the research environment, the number of electronic mail users must reach the critical mass that will "turn the business from a tough sell into a demand-pull industry," says *EMMS*'s Arnum. But the signs are all there: the suppliers of private electronic mail systems are emphasizing networking and connections to other vendors' systems; and the providers of public electronic mail services are finally recognizing the importance not only of connection to private systems but also of interoperation with each other. With the spread of PCs and desktop terminals, software is becoming easier to use and applications software is being developed to handle specific business needs. Already, some 5 million people have electronic mailboxes, both public and private, sending perhaps a couple of billion messages a year. And "there will be dramatic growth—60 billion electronic mail messages a year by the turn of the century," says Ulrich at Coopers and Lybrand. The critical mass, then, may not be far away.

Jeffrey Bairstow is a senior editor of HIGH TECHNOLOGY.

For further information see RESOURCES, *p.* 64.

ELECTRONIC MAIL HEADS TOWARD CRITICAL MASS

Electronic mail systems currently make up a $375 million market in North America, some 18% of a broader $2.1 billion electronic messaging market. This larger market encompasses a variety of technologies—including telex, facsimile, and voice mail equipment—used to transmit noninteractive messages over telecommunications links. By 1990, revenues from electronic mail systems are expected to top $1 billion, accounting for 29% of an overall $3.5 billion electronic messaging market, according to International Resource Development, a market research firm based in Norwalk, Conn.

Public services and private in-house systems represent the two major segments of the electronic mail market. About a million individuals subscribe to public services, according to Steve Glagow, manager of strategic business services at Walter Ulrich Consulting (Houston). Western Union (Saddle River, N.J.) controls 20–25% of this market; other participants include MCI (Washington, D.C.), GTE Telenet (Reston, Va.), British Telecom's Dialcom (Silver Spring, Md.), General Electric Information Services Co. (Rockville, Md.), and McDonnell Douglas (St. Louis). Message services are also offered as part of a broad array of communications options by CompuServe (Columbus, Ohio) and The Source (McLean, Va.).

Private systems, serving some 4 million people, are used by corporations, often in conjunction with integrated office automation systems. This market is dominated by the major computer manufacturers, including IBM (Armonk, N.Y.)—with about a 40% share—Digital Equipment Corp. (Merrimack, N.H.), Data General (Westboro, Mass.), Wang (Lowell, Mass.), and Hewlett-Packard (Palo Alto, Cal.).

"Computer-based message systems are spreading like wildfire," says Stephen Kirchoff, marketing manager for electronic mail at DEC. "Our company, for example, has grown from 2000 users to 50,000 on our own internal network in six years."

Two major factors lie behind such expansion. One is the increasing interconnectivity between public systems, private networks, and other forms of communications, such as telex and facsimile machines. "Connectivity is the key to the market's future," says J. Robert Harcharik, founder of MCI Mail and now an independent consultant in Washington, D.C, "because it brings to the industry a critical mass of individuals."

"Electronic mail systems can now transmit spreadsheet files, graphics, databases, and computer programs, as well as textual messages. Such added capabilities will help attract more customers."
Douglas Brackbill
Senior Marketing Manager, MCI

While connectivity is now developing on a case-by-base basis, such as the link between MCI Mail and CompuServe, the X.400 international electronic mail interconnection standard, set in 1984 by the Consultative Committee on International Telephony and Telegraphy, could open the door to worldwide networks. Public services based on X.400 are slated to begin operation this year in the U.K., France, and Germany; Dialcom and GTE say they plan to provide X.400 services in the United States. DEC has implemented this standard over its corporate electronic mail system, and systems have been demonstrated by IBM, Northern Telecom (Mississauga, Ont.), Sperry (Blue Bell, Penn.), and ICL (London), among others. "By 1990, there won't be a single vendor who hasn't incorporated X.400 into its products," says Robert Mealy, western regional sales manager at Sydney Development (Vancouver, B.C.); his firm makes a commercial X.400 system for use with computers from several manufacturers, including IBM, DEC, and Tandem.

The second factor stimulating industry growth is the development of niche markets and value-added services. For example, Speed > S, a small Minneapolis company, markets a message system that transmits nontextual files, such as spreadsheets and computer programs, between desktop machines more cheaply than can be done by some of the major public networks; customers include the IRS, which frequently sends changes in taxation programs to its field offices, and accounting firms. Magnatex International (Annapolis, Md.) has piggybacked a specialized network over Dialcom's system oriented to public-relations and advertising firms. And in the private network arena, Fisher-Innis (Naples, Fla.) and ADR (Princeton, N.J.) offer message systems for IBM mainframes.

"While the X.400 standard will allow systems to interconnect, it is not sufficient to make the market grow," concludes Tony Caplin, managing director of AirCall, a British telecommunications firm. "The future of the industry depends as much upon our ability to develop applications and to train users effectively." □—***Stephen A. Caswell***

RESOURCES

Information sources for topics covered in our feature articles

ELECTRONIC MAIL, P. 18

Contacts

Electronic Mail Association, 1919 Pennsylvania Ave., NW, Wash., DC 20006, (202) 293–7808.

Omnicom, 501 Church St. NE, Vienna, VA 22180, (703) 281–1135. Newsletter, courses, and videotapes on electronic mail.

Electronic Mail & Micro Systems, 6 Prowitt St., Norwalk, CT 06855, (203) 866–7800. Newsletter on electronic mail. $345/yr (biweekly).

References

"Getting the most out of electronic mail." Eddie Orton & Robin Nasatir. *Data Communications,* June 1986. Introduction to public electronic mail systems and vendors.

"Diamond: A multimedia message system built on a distributed architecture." Robert H. Thomas et al. *IEEE Computer,* Dec. 1985. Description of multimedia electronic mail system developed by BBN.

"The Information Lens: An intelligent system for information sharing in organizations." Thomas W. Malone et al. *Proceedings of the ACM Conference on Human Factors in Computing Systems,* Boston, April 1986. Description of intelligent filtering of electronic mail.

Proceedings of the Second International Symposium on Computer Message Systems, Wash., DC, Sept. 1985. North Holland (Amsterdam, Netherlands; in the U.S. c/o Elsevier Science Publishing, New York).

LITTLE SILICON VALLEYS, P. 24

State contacts

ALABAMA: Rick McClany, Ala. Development Office, State Capitol, Montgomery, AL 36130, (205) 263–0048.

ALASKA: Joe Yarzebinski, Economic Development, Department of Community Planning, P.O. Box 196650, Anchorage, AK 99519, (907) 264–4682.

ARKANSAS: Windell Adams, Arkansas Industrial Development Commission, 1 State Capitol Mall, Little Rock, AR 72201, (501) 371–2301.

ARIZONA: Marian Wallace, Arizona Department of Commerce, 1700 W. Washington St., 5th Fl., Phoenix, AZ 85007, (602) 255–5374.

CALIFORNIA: Jim Vaughn, Office of Business Development, Dept. of Commerce, 1120 L St., Suite 600, Sacramento, CA 95814, (916) 322–5665.

COLORADO: Stephan Andrade, Div. of Commerce & Development, 1625 Broadway, Suite 1710, Denver, CO 80203, (303) 892–3847.

CONNECTICUT: David Driver, Dept. of Economic Development, 210 Washington St., Hartford, CT 06106, (203) 566–4094.

DELAWARE: Donald C. Kane, Business Development Office, 99 Kings Hwy., P.O. Box 1401, Dover, DL 19901, (302) 536–4271.

FLORIDA: Ray Iannucci, Florida High Technology & Industry Council, Office of the Governor, The Capitol, Tallahassee, FL 32301, (904) 487–3134.

GEORGIA: Richard T. Meyer, Advanced Technology Development Ctr., 430 10th St., NW, Atlanta, GA 30318, (404) 894–3575.

HAWAII: Carl Swanholm, Dept. of Planning & Economic Development, P.O. Box 2359, Honolulu, HI 96804, (808) 548–8741.

IDAHO: Jay Engstrom, Div. of Economic Development, 108 State House, Boise, ID 83720, (208) 334–2470.

ILLINOIS: Norm Peterson, Commission on Science & Technology, 100 W. Randolph St., Suite 3–400, Chicago, IL 60601, (312) 917–3982.

INDIANA: Mark Akers, Industrial Development, Dept. of Commerce, 1 N. Capitol, Suite 700, Indianapolis, IN 46204, (317) 232–8888.

IOWA: Doug Getter, Dept. of Economic Development, 200 E. Grand Ave., Des Moines, IA 50309, (515) 281–3251.

KANSAS: Phil Bradford, Kans. Advanced Technology Commission, 400 W. 8th St., Suite 500, Topeka, KS 66603, (913) 296–5272.

KENTUCKY: Brad Richardson, Joe Brown, Industrial Development Div., Capitol Plaza Tower, Frankfort, KY 40601, (502) 564–7140.

LOUISIANA: Mary Jo Hanover, Business Development, La. Commerce Industry, Box 94185, Baton Rouge, LA 70804, (504) 342–5369.

MAINE: Henry Bourgeois, Maine Development Foundation, 1 Memorial Circle, Augusta, ME 04330, (207) 622–6345.

MARYLAND: Howard Aylesworth, Business & Devel. Office, Dept. of Economic Devel., 45 Calvert St., Annapolis, MD 21401, (301) 269–3176.

MASSACHUSETTS: Robert Crowley, Mass. Technology Development Corp., 84 State St., Suite 500, Boston, MA 02109, (617) 723–4920.

MICHIGAN: Jack Russell, Bureau of Innovation & Technology, Dept. of Commerce, 106 W. Allegan, Hollister Bldg., Rm. 212, Lansing, MI 48913, (517) 373–7411.

MINNESOTA: Jayne Khalifa, Office of Science & Technology, 900 American Ctr. Bldg., E. Kellogg Blvd., St. Paul, MN 55101, (612) 297–4368.

MISSISSIPPI: David Murphree, Research & Development Ctr., 3825 Ridgewood Rd., Jackson, MS 39211, (601) 982–6606.

MISSOURI: Allan Franklin, Advanced Technology Program, Dept. of Economic Development, 301 W. High St., Harry S. Truman Bldg., Jefferson City, MO 65101, (314) 751–3906.

MONTANA: Samuel T. Hubbard, Science & Technology Alliance, Dept. of Commerce, 1424 9th Ave., Helena, MT 59620, (406) 444–3923.

NEBRASKA: Netti Nelson, Div. of Telecommunications, Dept. of Economic Development, Box 94666, Lincoln, NE 68509, (402) 471–3341.

NEVADA: George Ormisten, Dept. of Economic Development, Capitol Complex, Carson City, NV 89710, (702) 885–4325.

NEW HAMPSHIRE: Paul Gilderson, Dept. of Research & Economic Development, P.O. Box 856, Concord, NH 03301, (603) 271–2343.

NEW JERSEY: Edward Cohen, Commission on Science & Technology, 122 W. State St., CN 832, Trenton, NJ 08625, (609) 984–1671.

NEW MEXICO: Patrick Rodriguez, Economic Development Div., Joseph Montoya Bldg., 1100 St. Francis Dr., Santa Fe, NM 87503, (505) 827–0272.

NEW YORK: Graham Jones, N.Y. Science & Technology Foundation, 99 Washington Ave., Albany, NY 11210, (518) 474–4349.

NORTH CAROLINA: Robert Brinkley, Industrial Development Div., Dept. of Commerce, 430 N. Salisbury St., Raleigh, NC 27611, (919) 733–4151.

NORTH DAKOTA: Charles Fine, Economic Devel. Commission, Liberty Memorial Bldg., Bismarck, ND 58505, (701) 224–4190.

OHIO: Larry McGeehan, Thomas Edison Program, 65 E. State St., Suite 200, Columbus, OH 43266-0330, (614) 466–3887.

OKLAHOMA: Carolyn Smith, Okla. Council of Science and Technology, Dept. of Commerce, 6601 Broadway Extension, Oklahoma City, OK 73116, (405) 521–2401.

OREGON: Peter Tryon, Mike Shadbolt, Economic Development Dept., 595 Cottage St., Salem, OR 97310, (503) 373–1200.

PENNSYLVANIA: Kant Rao, Technology & Policy Devel., Dept. of Commerce, 433 Forum Bldg., Harrisburg, PA 17120, (717) 783–5053.

RHODE ISLAND: Lewis Fazzano, Dept. of Economic Devel., 7 Jackson Way, Providence, RI 02903, (401) 277–2601.

SOUTH CAROLINA: Deb Wooley, S.C. Development Board, Box 927, Columbia, SC 29202, (803) 734–1400.

SOUTH DAKOTA: Dan Scott, Bureau of Industrial Development, Capitol Lake Plaza, Pierre, SD 57501, (605) 773–5032.

TENNESSEE: David Patterson, Tenn. Technology Foundation, P.O. Box 23184, Knoxville, TN 37933, (615) 966–2804.

TEXAS: David Brandon, Economic Development Commission, P.O. Box 12728, Capitol Station, 410 E. 5th St., Austin, TX 78711, (512) 472–5059.

UTAH: David Grant, Utah Economic Development, 6150 State Office Bldg., Salt Lake City, UT 84114, (801) 533–5325.

VERMONT: John Trethaway, Agency for Development & Community Affairs, 109 State St., Montpelier, VT 05602, (802) 828–3221.

VIRGINIA: David O'Donnell, Dept. of Industrial Development, 1000 Washington Bldg., Richmond, VA 23219, (804) 786–3791.

WASHINGTON: Alan Harger, Council for Technology Advancement, Dept. of Commerce & Economic Development, 101 General Admin. Bldg., Olympia, WA 98504, (206) 753–3065.

WEST VIRGINIA: Alex McLaughlin, Research & Strategic Planning, Governor's Office of Community & Industrial Development, Capitol Complex, Charleston, WV 25305, (304) 348–2234.

WISCONSIN: Ralph Wegenke, Dept. of Development, Box 7970, Madison, WI 53707, (608) 266–9869.