UNCLASSIFIED

AD NUMBER
AD855579
NEW LIMITATION CHANGE
TO Approved for public release, distribution unlimited
FROM Distribution authorized to U.S. Gov't. agencies and their contractors; Administrative/Operational Use; 01 MAR 1969. Other requests shall be referred to Rome Air Development Center, Griffiss AFB, NY.
AUTHORITY
RADC ltr, 17 Sep 1971

THIS PAGE IS UNCLASSIFIED

AD855579

RADC-TR-69-98
8 March 1969

STUDY FOR THE DEVELOPMENT OF COMPUTER AUGMENTED MANAGEMENT TECHNIQUES

Douglas C. Engelbart
William K. English
David A. Evans

Contractor: Stanford Research Institute
Contract Number: F30602-68-C-0286
Effective Date of Contract: 10 April 1968
Contract Expiration Date: 10 April 1970
Amount of Contract: $1,515,222
Program Code Number: 7D30 and 9D30

Principal Investigator: D. C. Engelbart
Phone: 415 326-2220

Project Engineer: D. F. Bergstrom
Phone: 315 330-2600

Sponsored By
Advanced Research Projects Agency
ARPA Order No. 967

This document is subject to special export controls and each transmittal to foreign governments, foreign nationals or representatives thereto may be made only with prior approval of RADC (EMIIH), GAFB, N.Y. 13440.

Rome Air Development Center
Air Force Systems Command
Griffiss Air Force Base, New York

D D C
JUL 30 1969

73

ACCESSION for
CFSTI WHITE SECTION ☐
DDC BUFF SECTION ☑
UNANNOUNCED ☐
JUSTIFICATION .....
BY .....
DISTRIBUTION/AVAILABILITY CODES
DIST. AVAIL. and/or SPECIAL
2

When US Government drawings, specifications, or other data are used for any purpose other than a definitely related government procurement operation, the government thereby incurs no responsibility nor any obligation whatsoever; and the fact that the government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data is not to be regarded, by implication or otherwise, as in any manner licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

If this copy is not needed, return to RADC (EMIIH), GAFB, N.Y. 13440.

STUDY FOR THE DEVELOPMENT OF COMPUTER AUGMENTED MANAGEMENT TECHNIQUES

Douglas C. Engelbart
William K. English
David A. Evans

Stanford Research Institute

This document is subject to special export controls and each transmittal to foreign governments, foreign nationals or representatives thereto may be made only with prior approval of RADC (EMIIH), GAFB, N.Y. 13440.

This research was supported by the Advanced Research Projects Agency of the Department of Defense and was monitored by D.F. Bergstrom RADC (EMIIH), GAFB, N.Y. 13440 under Contract No. F30602-68-C-0286.

ON-LINE WORK STATION IN AN EXPERIMENTAL OFFICE
One of the authors, W. K. English, is in control chair. His right hand guides "mouse" (cursor device) while his left operates 5-finger keyset. Screen shows part of a file containing mixed text and graphics. See Sections II-E and III-C-2 for descriptions of system and experimental office.

ABSTRACT

This report deals with certain aspects of a continuing research effort on direct augmentation of human intellectual effectiveness by means of full-time use of special on-line computer aids. Areas of special importance in this report are development and maintenance of a computer facility centered around a time-sharing SDS 940 and the development of on-line computer aids for management.

CONTENTS

ABSTRACT.....v

LIST OF ILLUSTRATIONS.....ix

LIST OF TABLES.....xi

NOTE ON CONTRIBUTORS.....xiii

PUBLICATION REVIEW STATEMENT.....xiii

I INTRODUCTION.....1

II THE AUGMENTED HUMAN INTELLECT RESEARCH CENTER.....3

III MANAGEMENT SYSTEMS RESEARCH ACTIVITY.....13

IV FACILITY DEVELOPMENT AND MAINTENANCE.....25

V GENERAL DEVELOPMENTS OF RELEVANCE.....43

GLOSSARY.....47

REFERENCES.....49

BIBLIOGRAPHY.....51

Appendix A: AHI PARTICIPATION IN THE FJCC.....53

Appendix B: PROPOSALS FOR EXPANSION.....59

Appendix C: THE NLS VECTOR PACKAGE.....63

ILLUSTRATIONS

Frontispiece	USER'S WORK STATION IN AN EXPERIMENTAL OFFICE ENVIRONMENT	iii
Fig. 1	LOAD FACTOR AS A FUNCTION OF NUMBER OF USERS	33
Fig. 2	INPUT-FEEDBACK RESPONSE TIMES	34
Fig. 3	SHORT-LINE DISPLAY-CREATION RESPONSE TIMES	35
Fig. 4	LONG-LINE DISPLAY-CREATION RESPONSE TIMES	36
Fig. 5	DISTRIBUTION OF INTERVALS BETWEEN DISPLAY-CREATION ACTIONS BY INDIVIDUALS	37
Fig. 6	DISTRIBUTION OF INTERVALS BETWEEN CHARACTER-INPUT ACTIONS BY INDIVIDUALS	38

TABLES

Table I LOAD FACTOR AS A FUNCTION OF NUMBER OF USERS.....39

Table II LOGS AND CORRESPONDING TIMES.....39

Table III INTERVAL DISTRIBUTION OF DISPLAY CREATION BY INDIVIDUALS.....40

Table IV INTERVAL DISTRIBUTION OF CHARACTER INPUT BY INDIVIDUALS.....40

NOTE ON CONTRIBUTORS

The persons directly responsible for this report are D. C. Engelbart, W. K. English, and D. A. Evans. Overall credit for the work reported should go to the staff of the Augmented Human Intellect Research Center as a whole.

In particular, J. F. Rulifson and D. I. Andrews are responsible for the study of system loads; W. H. Paxton for the Vector Package and the Keyword Information-Retrieval System; F. K. Tomlin for the film output system; and D. Casseres for some of the background documentation appearing in this report.

PUBLICATION REVIEW

This technical report has been reviewed and is approved

RADC Project Engineer

I INTRODUCTION

This interim technical report covers work performed in the first year of Contract F30602-68-C-0286, SRI Project 7101, for the period 8 February 1968 through 8 February 1969.

The work is reported in three major sections. The first (Section II) is a general account of activities in progress in the AHIRC, with a brief description of the On-Line System (NLS).

The other two sections (Sections III and IV) correspond to the main division of work supported under this contract.

Section III reports on the Management Systems Research activity; Section IV reports on the Facility Development activity.

Management Systems Research is concerned with the design, development, and implementation of on-line computer aids for management, and with the study of an organization of on-line workers.

The Facility Development activity is concerned with the development and operation of a timeshared computer facility to support this on-line community.

Both the Management Systems Research activity and the Facility Development activity are part of the system of activities that now constitutes the Augmented Human Intellect Research Center (AHIRC) at Stanford Research Institute (SRI).

The role and value of work reported in this document cannot be fully understood without a prior understanding of the overall objectives, strategy and current research of the AHIRC. These are reported briefly in the section following this introduction.

The work reported in the section on Management Systems Research covers the following areas: On-Line System (NLS) developments of greatest relevance, the study of an organization of on-line workers, development of graphs and representations, and specifications for an on-line calculator.

Areas reported for the Facility Development activity are as follows: status of facility (time-sharing system, hardware, and computer-generated sound), the study of system loads, and some considerations for the reliability of data.

I INTRODUCTION

This report includes a section on general developments of relevance to the AHI research center. These include the development of hard-copy output, the proposal for increase in scope, AHIRC participation in the Fall Joint Computer Conference, and recent innovations in the On-Line System (NLS).

Note: The format of this report reflects the use of on-line computer aids to produce it. The format is explained in Section II-E-2.

II THE AUGMENTED HUMAN INTELLECT RESEARCH CENTER

A. Augmentation Systems

An "augmentation system," in the larger sense, is the whole system of concepts, representational forms, procedures, conventions, attitudes, software, hardware, etc. that can be developed and employed by man to augment his own intellectual capabilities.

In the AHIRC, the central technological element in an augmentation system is a time-shared computer facility with graphic displays, providing a community of research workers with highly interactive on-line computer aid.

B. Objectives

The basic objective of the AHIRC is to develop better augmentation systems and to develop a system-oriented discipline that embodies the principles and techniques to be used in the design and development of such systems.

The objectives, strategy, and operation of the Augmented Human Intellect Research Center are described in more detail in several other publications (see Refs. 1, 2, 3, and 4).

C. Organization

The AHIRC currently has a staff of 19, including 13 professionals. Plans are being implemented to increase the staff to a total of 29 (see comments on the recent proposal for increase in scope, Appendix B).

The Center's professionals are currently served by 6 on-line graphic display terminals served by an SDS 940 time-shared computer. Details of this facility are reported in Section IV.

Within the Center, research is focused on seven interrelated activities:

(1) Management Systems Research

(2) Service System Hardware Development

(3) Service System Hardware Maintenance

(4) Service System Software Development

(5) User System Research

(6) Connection of the AHIRC computer facility to the ARPA Network

(7) Network Information Center.

D. Research Strategy: Bootstrapping

In order to approach these objectives, the Center has adopted an unusual research strategy called "bootstrapping." Research products developed in the Center are immediately put to work within the Center.

Researchers in the Center are concerned with designing, building, and using man-computer augmentation systems that will improve their ability to design, build, and use better augmentation systems.

For example, NLS, an interactive aid used to compose, study, and modify text and graphic information, is available to system programmers (and others) at display terminals in the Center. A system programmer may thus use NLS to more effectively write, modify, edit, and debug new programs that can, in some cases, produce an improved NLS, and thus provide him with an improved system for writing, editing, and debugging more programs.

The bootstrap strategy is designed to capture the maximum amount of feedback from the Center's own research work, in order to make it possible for the Center to do increasingly more effective research in the field of man-computer systems.

This strategy leads to evolutionary growth, with an intimate mixture of theory and practice in the development of man-computer augmentation systems.

Bootstrapping applies to the Management Systems Research activity, as it does to all other activities in the Center.

We are actively engaged in using within the Center the management aids we have developed. This provides a realistic testing ground for these new systems and is the basis for their continued development, improvement, and evaluation.

As better management aids are developed, presumably the Center's actual management will greatly improve in efficiency and quality, making possible the coordination and control of increasingly complex research activity.

This second characteristic of Management Systems Research in the AHIRC is an excellent example of the more subtle and

powerful implications of the bootstrap strategy; these implications operate at many "unseen" levels, generated by virtually every research activity in the Center.

E. The On-Line System (NLS)

Among the major research products of the AHIRC is NLS, the On-Line System. The development of NLS is not directly sponsored by RADC, but specific research on management systems, which is sponsored by RADC, depends upon NLS as the basic infrastructure for its development. Likewise, Management Systems Research will contribute to the development of special features in NLS.

NLS is continually evolving. The following is a brief outline of NLS as it exists today, designed to familiarize the reader with the general architecture and nature of NLS.

NLS, as currently implemented, is essentially a highly sophisticated text-manipulation system (with some graphics capabilities) oriented primarily toward on-line use; i.e., it is not primarily oriented toward production of hard copy, although fairly sophisticated hard-copy formatting and output are included in the system.

NLS is intended to be used on a regular, more or less full-time basis in a time-sharing environment, by users who are not necessarily computer professionals. The users are, however, assumed to be "trained" as opposed to "naive." Thus the system is not designed for extreme simplicity, nor for self-explanatory features, nor for compatibility with "normal" working procedures.

Rather, it is assumed that the user has spent considerable time in learning the operation of the system; that he uses it for a major portion of his work; and that he is consequently willing to adapt his working procedures to exploit the possibilities of full-time, interactive computer assistance.

Thus the practices and techniques developed by users for exploiting NLS are as much a subject of research interest as the development of NLS itself.

1. Work-Station Console

The user sits at a console whose main elements are a display screen, a typewriter keyboard, a cursor device called the "mouse," and a set of five keys operated by the left hand, called the "keyset" (see Frontispiece).

The screen is used for displaying text, in various formats.

The top portion of the screen (approximately 1/5 of the total area) is reserved for feedback information of various kinds: the name of the user command mode currently in effect, a "register" area used for various kinds of feedback, an "echo register" which displays the last six characters typed by the user, and other items which are explained below.

The keyboard closely resembles a conventional typewriter keyboard, with a few extra keys for special characters and control functions. It is used for typing text as content for a file and for specifying commands, which are given as two- or three-character mnemonics.

The mouse is a roughly box-shaped object, about four inches on its longest side, which is moved by the right hand. It is mounted on wheels, and rolls on any flat surface. The wheels drive potentiometers which are read by an A/D converter, and the system causes a tracking spot ("bug") to move on the screen in correspondence to the motion of the mouse.

The user specifies locations in the displayed text by pointing with the mouse/bug combination. This eliminates the need for specifying a location by entering a code of some kind. Use of the mouse is very easily learned and soon becomes unconscious.

On top of the mouse are three special control buttons, whose uses are described below.

The keyset has one key for each finger of the left hand. The keys are struck in combinations called "chords," and each chord corresponds to a character or combination of characters from the keyboard. There are 31 possible chords; beyond this, two of the buttons on the mouse may be used to control the "case" of the keyset, giving alternate meanings to each chord. There are four possible cases, for a total of 124 possible combinations.

A simple binary code is used: it has proved remarkably easy to learn. Two or three hours' practice are usually sufficient to learn the most commonly used chords and to develop reasonable speed.

The keyset was developed to increase the user's speed and smoothness in operating NLS. It was found that users normally keep the right hand on the mouse, because the great majority of command operations involve a pointing

action; efficient use of the keyboard, however, requires the use of both hands, and shifting the right hand (and the user's attention) to the keyboard is distracting and annoying if it must be done for each two- or three-letter command mnemonic.

Use of the keyset permits the user to keep his right hand on the mouse and his left on the keyset, reverting to the keyboard only for entry of long strings of text (typically five or more characters).

Originally, the keyset exactly duplicated the keyboard in function; in the development of NLS, however, certain control functions have been made two-stroke operations from the keyset where they would be three- or four-stroke operations from the keyboard. Nevertheless, it is still possible to operate all of the features of NLS without using the keyset; thus the beginner may defer learning the keyset code until he has gained some degree of mastery over the rest of the system.

2. Structured Text

"Text" is used here as a very general term. A "file" of text (corresponding roughly to a "document" in hard copy) may consist of English or some other natural language, numerical data, computer-program statements, or anything else that can be expressed as a structure of character strings. Simple line drawings can also be included in a file.

All text handled by NLS is in "structured-statement" form. This special format is simply a hierarchical arrangement of "statements," resembling a conventional "outline" form.

Each statement in a file may be considered to possess a "statement number," which shows its position and level in the structure. Thus the first statement in a file is Statement 1; its first substatement is 1a, and its next substatement is 1b; the next statement at the same level as the first is Statement 2; and so forth. Statement numbers have been suppressed in printing out most of this report, but they are printed out for the remainder of this section as an example.

1e5b2 A statement is simply a string of text, of any length; this serves as the basic unit in the construction of the hierarchy. In English text, statements are normally equivalent to paragraphs, section and subsection headings, or items in a list. In other types of text, statements may

be data items, program statements, etc.

1e5b2a Each paragraph and heading in this report is an NLS statement. Each statement is indented according to its "level" in the hierarchy; this paragraph is a substatement of the one above, which is in turn a substatement of another statement. A statement may have any number of substatements, and the overall structure may have any number of levels.

1e5c Note that when a user creates a file, he may let all of his statements be first-level ones, i.e. 1, 2, 3, etc. In this case he will not have to consider a hierarchical structure but simply a linear list, as is found in conventional text.

1e5c1 However, many of the features of NLS are oriented to make use of hierarchy, and the benefits of these features are lost if hierarchy is not exploited.

3. Use of the System

Text manipulation is considered to involve three basic types of activity by the user: composition, study, and modification. In practice, the three activities are so intermingled as to be indistinguishable.

a. Composition

Composition is simply the creation of new text material as content for a file.

In the simplest case, the user gives the command "Insert Statement" by typing "is". He then points (with the mouse) to an existing statement; the system displays a new statement number which is the logical successor, at the same level, as the statement pointed to. The user may change the level of this number upward by typing a "u" or downward by typing a "d".

Note: Even if no previous statement has been created, the system displays a "dummy" statement at the top of the text-display area, and the user points to this dummy.

The user then types the text of the new statement from the keyboard. On the screen, the top part of the text-display area is cleared and characters are displayed here as they are typed. When the statement is finished, the user hits a CA (command accept) button on the keyboard or mouse, and the system recreates the display with the new statement

following the one that was pointed to.

New material may also be added to existing statements by means of commands such as Insert Word, Insert Text, and others. Properly speaking, these operations are modification rather than composition, and are discussed below.

Simple line drawings may be composed and added to the file by means of the "vector package." This is discussed in Section III-D-1.

b. Study

The study capabilities of NLS constitute its most powerful and unusual features. The following is only a brief, condensed description of the operations that are possible.

Jumping: NLS files may, of course, contain a great deal more text than can be displayed on the screen, just as a document may contain more than one page of text. An NLS file is thought of as a long "scroll." The process of moving from one point in the scroll to another, which corresponds to turning pages in hard copy, is called "jumping." There is a very large family of Jump commands.

The basic Jump command is Jump to Identity. The user specifies it by entering "ji", and then points to some statement with the mouse. The selected statement is moved to the top of the screen, as if the scroll had been rolled forward.

Most of the Jump commands reference the hierarchical structure of the text. Thus Jump to Successor brings to the top of the display the next statement at the same level as the selected statement; Jump to Predecessor does the reverse; Jump to Up starts the display with the statement of which the selected statement is a substatement, and so forth.

The Jump to Name command uses a different way of addressing statements. If the first word of any statement is enclosed in parentheses, the system will recognize it as the "name" of the statement. Then, if this word appears somewhere else in the text, the user may jump to the named statement by pointing to the occurrence of the name, or by typing the name.

This provides a cross-referencing capability which is

very smooth and flexible; the command Jump to Return will always restore the previous display, so that the user may follow name references without losing his place.

It is also possible to jump to a statement by typing its statement number.

View Control: If a file is long, it may be impossible for the user to orient himself to its content and structure or to find specific sections by jumping through it. The principal solution to this problem is provided by level control and line truncation.

Level control permits the user to specify some number of levels; the system will then display only statements of the specified level or higher. Thus if three levels are specified, only first-, second-, and third-level statements are displayed.

Line truncation permits specification of how many lines of each statement are to be displayed. Thus if one line is specified, only the first line of each statement will be displayed.

Common usage is to use the first two or three levels in a file as headings describing the material contained under each heading in the form of substatements. Thus the user may start by looking at a display showing only the first-level statements in the file, one line of each. This amounts to a table of contents.

He may then select one of these statements and jump to it, specifying one more level. He will then see more details of the content of that part of the file. This process of "expanding the view" may be repeated until the user has found what he is looking for, at which point he may specify a full display of the text.

Users soon develop a habit of structuring files in such a way that this process will work well. As it happens, such a structure is usually a good, logical arrangement of the material, reflecting the relationships inherent in the content.

The level and truncation controls are designed so that the necessary specifications may be made with only one or two strokes of the keyboard or keyset. These controls are only the most important of a large set of view-control parameters

called "VIEWSPECs." Other VIEWSPECs control a number of special NLS features affecting the display format.

Content Analysis: The NLS content analyzer permits automatic searching of a file for statements satisfying some content pattern specified by the user. The pattern is written in a special language as part of the file text.

This feature is discussed in greater detail in Section III-B-2.

Link Jumping: A "link" is a string of text, occurring in an ordinary file statement, that indicates a cross-reference of some kind. It may refer to another statement in the file, or to a statement in some other file, possibly belonging to another NLS user. The text of the link is both human-readable and machine-readable, and the command Jump to Link permits the user to point to the link with the mouse and immediately see the material referred to.

This feature is discussed in greater detail in Section III-B-3.

c. Modification

A large repertoire of editing commands is provided for modification of files. The basic functions are Insert, Delete, Replace, Move, and Copy.

These functions operate upon various kinds of text entities. Within statements, they may operate upon single characters, words, and arbitrary strings of text defined by pointing to the first and last characters.

This set of commands is not restricted to operation within one statement at a time; for example, a word may be moved or copied from one statement to another.

The editing functioz also operate at the structural level, taking statements or sets of statements as operands. A number of special entities have been defined for this purpose: for example, a "branch" consists of some specified statement, plus all of its substatements, plus all of their substatements, etc. A branch can be deleted, moved to a new position in the structure, etc.

As noted above, the modification activity tends to merge, in practice, with study and composition.

4. Summary

It must be noted that NLS is not a system designed for immediate use by any sort of "general public," but an experimental system designed to evolve under constant use by a group of persons working on the development of computer aids to human intellectual processes. It is for this reason, for example, that NLS is not really a text-editing system oriented toward hard-copy production, but rather something simultaneously more general and more specialized.

It is in the process of manipulating a file -- studying it, making modifications, adding new material as an integrated process lasting for minutes or hours at a time and having a continuity extending for days, weeks, or even years -- that the real benefit of NLS appears.

An NLS file tends to become an evolving entity, subject to constant modification, updating, and reevaluation. Its development may have no clearly defined endpoint. It may cease to exist as a file by being incorporated in another file, or it may eventually be abandoned; however, it will probably never be "finished" in the usual sense of the word.

Continuous use of NLS to store ideas, study them, relate them structurally, and cross-reference them results in a superior organization of ideas and a greater ability to manipulate them further for special purposes, as the need arises -- whether the "ideas" are expressed as natural language, as data, as programming, or as graphic information.

III MANAGEMENT SYSTEMS RESEARCH ACTIVITY

A. General

The period February 1968 through February 1969 has seen the firm establishment of the AHIRC on-line community. The work involved in making this community operable proved to be more time-consuming than estimated at the beginning of this contract, but neverthless, reasonably reliable and useful on-line service was made available to the Center's staff by mid-July 1968.

Initially this service was quite limited. NLS (the On-Line System) was made available at display terminals, in an initial version consisting only of the basic editing and file manipulation commands.

File storage was provided on the random-access drum and on magnetic tape.

However, NLS and associated hardware service improved rapidly. By August 1968, a more reliable and comprehensive On-Line system was available to users.

The Bryant disc system was accepted on 22 August. This greatly enhanced the value of the system, providing users with disc storage for NLS files in place of tape storage.

By September 1968, NLS was operating very well, and console-use time increased markedly. About this time NLS surpassed the power of the old NLTS (On-Line Text System) implemented on the CDC 3100, partly supported by RADC under Contract AF 30(602)-4103.

Progress of Management Systems Research has been largely determined and constrained by the development of the the On-Line System in general, the development of the hardware facility, and the growth of the on-line community as a whole.

As a result of delays in making the community operational, the Management Systems Research activity is only now gaining momentum, at the end of the period covered by this report. With the present sophistication of NLS, rapid progress is now planned in the development and implementation of on-line aids to management.

There is no doubt that the delay has been worthwhile. There now exists an evolving infrastructure of on-line aids tha will provide a significantly richer framework to support further research on Management Systems. This first year should be regarded as a "tooling-up" phase as far as the Management Systems Research activity is concerned.

III MANAGEMENT SYSTEMS RESEARCH ACTIVITY

Progress in Management Systems Research is described in the following sections.

B. Relevant NLS Developments

Among the NLS features developed in the first year of this contract (but not directly funded by RADC) that have the greatest bearing on Management Systems Research and the greatest value as existing on-line aids to management are the Keyword Retrieval System, the Content Analyzer, the Link Jump feature, and the Viewchange system. These are described in summary form below.

These features will play a key role in the development of future on-line aids specifically tailored for management functions. They should be regarded as among the basic building blocks for a sophisticated system of management aids.

1. Keyword Retrieval System

The keyword information-retrieval system is essentially a method of reordering statements in an NLS file in accordance with some selected and weighted subset of "keywords."

There are a great many possible uses for this operation; the most important application that has been evolved to date is to use the keyword system on a "catalog" file which lists other NLS files and includes a system of keywords applying to the listed files.

A catalog file for use with the keyword system consists of two parts:

A list of keywords used, with information as to their meanings

A list of files referenced, with information as to their contents, machine-readable lists of applicable keywords, and machine-executable "link" text for retrieving the actual files.

To use the catalog, the user examines the keyword listing and selects (with the mouse) any keywords that relate to desired types of information. He may also assign weights to the keywords, on a scale from 1 to 10.

Upon execution, the system assigns "scores" to the entries in the file listing, according to the number and weights of keywords applying to each entry.

It then displays a list of entries with nonzero scores, in order of increasing score. This constitutes an ordered list of files containing the types of information desired by the user.

The user may now retrieve individual files with the Jump to Link feature (described in Section III-B-3).

Detailed information on the operation of the information-retrieval system is given in Appendix F of Quarterly Progress Report 1 under the present contract (Ref. 1).

2. Content Analyzer

The Content Analyzer feature of NLS is a highly flexible system allowing the user to specify a particular "pattern" of content, and then cause display of only those statements in a file that satisfy that pattern.

The pattern specification is written into the file itself, as text, in a special high-level language. The pattern may be simple -- e.g. it may call for the occurrence anywhere in a statement of some particular word -- or it may be highly complex, e.g. it may involve the occurrence of several words or characters in some special relationship within the statement.

The specification of simple patterns is simple and easy; complex patterns may require fairly intricate formulation.

Stored with every statement in a file is information giving the initials of the last person to change the text of the statement and the date and time on which this occurred. This information may be interrogated by the content analyzer: thus the user may write a pattern whose meaning is "display statements written or changed by anyone except me since the beginning of this week."

Detailed information on the operation of the content analyzer is given in Appendix E of Quarterly Progress Report 1 under the present contract (Ref. 1).

3. Link Jumps

By means of "links," it is now possible for the user to move about in several different files as easily as he formerly moved in a single file.

All files used in this fashion must be stored on one of the rapid-access devices (disc or drum). The current configuration of the system limits the number of files that can be so stored,

but plans for the near future include the ability to store many files in this fashion.

A link is a string of text specifying a particular file, a location in that file, and a set of viewing parameters to be used in displaying the file initially. The format of the link is (USERNAME,FILENAME,LOCATION:VIEWSPECS).

The user name is omitted if the referenced file belongs to the same user as the current file.

The file name is omitted if the link refers to a location in the current file.

If the location is omitted, the beginning of the file is assumed.

If the VIEWSPECs are omitted, the current VIEWSPECs are assumed.

This text string, which is readable by the user or by the system, is inserted in the file text in the same way that a cross-reference would be used in hard copy. Note that it may refer to another file or to another location in the current file.

When a user encounters the link, he may follow it simply by giving the command "Jump to Link" and pointing to the link with the cursor. The referenced material is immediately displayed; the user may then return to the previous display or he may follow another link if there is one.

Detailed information on link jumps may be found in Appendix D of Quarterly Progress Report 1 under the present contract (Ref. 1).

4. Viewchange

A large set of user commands has been implemented to give the NLS user the power to reformat his display. The various feedback entities normally displayed at the top of the screen may be individually moved, modified, or deleted, as may the main text area of the display.

Any of these display entities may be suppressed completely, or moved to a different part of the screen, or displayed in larger or smaller characters than normal. They may also be displayed in italic, underlined, or flashing characters.

The command feedback line may also be displayed in abbreviated form, showing only the initial letters of the words normally displayed.

In the main text area, the line length and the number of lines may be adjusted.

The two characters used for displaying the cursor spot ("bug") may be changed.

Once the user is satisfied with a new display format, he may cause the various display parameters that he has set to be stored as part of his file. At any future time he may then retrieve them with a user command and put them into effect.

One purpose of this system is to permit reformating of the display for use in video communication. Thus a user may clear some area of his screen (e.g. top or bottom half, right or left half, or one corner) to be used for superimposing a video image of his own face, another user's face, another user's display, or some other image.

Experimentation with these techniques is still at an early stage, but some of possibilities were explored quite extensively in the presentation for the 1968 Fall Joint Computer Conference, where a large screen with the speaker's console display projected on it was used as a primary means of communication to the audience (see Appendix A).

Detailed information on viewchanging may be found in Appendix C of Quarterly Progress Report 1 under the present contract (Ref. 1).

C. Study of an Organization of On-Line W orkers

1. General

The fact that a group of research workers now regularly make use of our augmentation systems as a part of their day-to-day work has made it possible to begin the part of the Management Systems Research activity concerned with studying an organization made up of on-line workers; this has been a unique opportunity.

Work in this area to date has been confined to informal observation and examination, as well as planning for substantive research to be performed in the coming year.

Already some extremely interesting features of an on-line

community are evident. These features have not yet been subjected to formal study or analysis; they are (at the present time) the product only of general observation and consensus among researchers in the Center.

These features include the formation of "dynamic" cliques of staff members (i.e. groups whose members change from time to time without the group losing its identity), the high emotional level of day-to-day life in the organization, the spontaneity of new ideas, the newly emerging procedures whereby important innovations "surface", and the importance of the organization's physical environment.

Only fragmentary remarks can be made at this time about the first four of these features, and discussion of these will be deferred to the final report. The fifth feature -- the importance of the organization's physical space -- is more tangible. Since it is of immediate concern to the Center, it is discussed in some detail below (Section III-C-2).

At this stage it is impossible to distinguish clearly between the influence of features peculiar to the AHIRC (e.g. personalities of researchers) and the influence of more general forces related to the nature and usage of the augmentation systems that are at the focus of the Center's research.

At best we can say that these influences work together to account for the nature of the AHIRC as an on-line organization, and that similar forces will act in similarly interrelated ways in all future on-line organizations.

The interaction of these forces is present in all types of organizations. Our own experience indicates that the nature of augmentation systems being developed in the Center intensifies this interaction and places new emphasis on the relationships between idiosyncrasies of individuals and the form of the organization of people, machines and systems that constitutes the environment in which these individuals work. The problem within the framework of on-line organizations can thus be assumed to be distinct from that in most conventional organizations, and thus warrants study by itself.

We can also say that any system to aid in the management of such an organization must recognize all dimensions of the organization -- the emotional and personal as well as the technological.

Because of AHIRC's current lead in the development of interactive on-line aids that are being employed as part of the day-to-day life of intellectual workers, it is reasonable to consider the AHIRC as a prototype on-line organization, and to report our understanding of the state and character of this organization as a part of formal and informal studies of on-line organizations. Such information is particularly relevant in the development of on-line aids for management of such an organization.

2. Physical Environment for an On-Line Organization

The nature of NLS supports team efforts and establishes a need to make it possible, as a regular procedure, for groups of individuals to get together for quick conferences around a single console, then for each individual to return to a single console.

Clearly, the conventional cubicle office arrangement does not support this type of personal interaction.

With this sort of problem in mind, the AHI Research Center has been fortunate to interest Herman Miller Research Corporation in our situation, and we have been experimenting with their very flexible forms of walls and work spaces to support the type of interaction and privacy/exposure relationships that are needed in an on-line community.

A small experimental work area has been set up at SRI. This work space incorporates several innovations judged by AHIRC staff to be important to their physical environment (see Frontispiece). These innovations are as follows:

A comfortable "control chair," incorporating the controls for operating the display terminal, has been developed. This unit is independent of the display itself, allowing the user to position himself freely with respect to the screen.

A molded plastic control panel mounted across the arms of the chair contains a keyset, a keyboard, a mouse, and several switches to control special display features. The panel swings to one side to allow the user to sit down or get up.

With this control chair and panel, an individual can achieve far greater flexibility in using the system, especially if he is conferencing with several other people, using the On-Line System as the focus of the group's attention. The

basic advantage of this configuration over a conventional console is that the controlling user does not block the view of other parties and all parties can face each other in a satisfactory conferencing arrangement.

The plastic control panel can be detached from the chair, and can thus be handed from person to person in a group to give system control to the most appropriate party in a conference at the time it is needed. Alternatively, the panel can be placed on a stand-up work surface to give the on-line user variety in his work posture.

This is of considerable importance for individuals who use an on-line system for long periods of time without a break. This type of usage is common in the AHI Research Center, and will be common in on-line communities of the future.

Another innovation is the provision of wall panels that can be easily moved, rearranged, removed, or built upon to provide the type of acoustic and visual privacy required by a user (or users) for a given purpose.

The components making this possible are standard Herman Miller Action Office II wall components, recently released on the market.

We are only beginning to understand the operation of an on-line community. Our present limited experience with the use of physical space as an integral part of our augmentation systems suggests that space is a very important variable, and one over which we have a great deal of control.

In the future we expect continued formal and informal research on physical space, and the initiation of serious research into other aspects on the behavior and management of the on-line community.

Within the next months, the AHI Research Center will move to a new location in the building we presently occupy. This new location will permit further experiments with spatial relationships.

The basic configuration of this new location will be researcher's private offices opening onto a large central work area, in which most of our on-line terminals will be placed. We plan to make maximum use of the flexible type of physical environment described above to provide a working space that supports our unique requirements.

D. Development of Representations as Management Aids

An important part of Management Systems Research is the development of graphs, plots, and other representations that will be useful as part of a system of on-line management aids.

As with other Management Systems Research, this work is only now gaining momentum. During the first year, a first-stage graphics capability called the Vector Package was developed, partly funded by this contract. Using the Vector Package, an on-line user can make simple line drawings, graphs, etc. as part of a regular NLS file.

The Vector Package is completely passive — no computational processes can be used to generate graphs. It was implemented to investigate the way non-text information can be integrated wth NLS, and has served this purpose extremely well.

The development and use of the Vector Package has contributed to the specification of a second-generation graphics package. This proposed system includes features especially tailored for on-line management aid. The Vector Package and specifications for the new graphics system are described in some detail below.

1. Vector Package

The Vector Package is a first-stage graphics system operating as an integral part of NLS.

Every drawing created with the vector package is associated with a statement in an NLS file. The drawing consists of straight lines (vectors) and labels (text, with the same character set as NLS itself).

a. Vectors

The Insert Vector command allows the user to define two endpoints with the mouse; a vector is then automatically drawn between these endpoints. The second endpoint may be used as the first endpoint of the next vector, or the next vector may start elsewhere.

After a vector has been created with this command, it may be moved, translated, deleted, or projected to the vertical or horizontal. Each of these operations makes use of a selection made with the mouse. The user points to the vector and hits the Command Accept (CA) button; the system then marks the ends of the vector with an O and an X.

In the Delete Vector command, the vector is immediately deleted upon execution of the command.

In the Move Vector command, the end marked X will move to a new position selected with the mouse and the end marked O remains fixed. In the Translate Vector command, the end marked X again moves to a new position selected with the mouse, but the end marked O also moves in such a way as to preserve the length and direction of the vector.

In the Horizontal and Vertical commands, the end marked X moves vertically until the vector is horizontal, or horizontally until the vector is vertical.

b. Labels

In the Insert Label command, the user types the text of a label and this text is then "attached" to the cursor. The user moves it to the desired location and hits CA; the label is then fixed. In the Delete Label command, the user simply points to a label and it is deleted on CA; to move a label, the user may cause it to be "attached" to the cursor again and move it to any desired position.

If a label in a drawing also occurs as the "name" (parenthesized first word) of a statement in the file, the label may be used as a machine-executable cross-reference link. (See Section II-E-3-b.)

c. Other Commands

The user may move the entire drawing (set of existing vectors and labels) relative to the associated statement text, or he may delete the entire drawing and start over. A grid of dots may be superimposed for use as a guide in drawing vectors and locating labels.

A portion of the NLS User's Guide relating to the vector package is given as Appendix C.

2. Specifications for the New Graphics Package

Specifications for the second-generation NLS graphics package are being formulated at present.

Final specifications have not yet been determined, but the following characterstics should be included in the new system:

Automatic generation of graphs, charts, and other

representations as a part of an NLS file, from a data base and calculation package accessible from directly within NLS.

Direct compatibility between picture entities and regular NLS statements

Picture editing operations (including move, delete, replace, etc.) on individual picture entities and groups of entities.

Provision of a library of picture entities (e.g. squares, circles, rectangles, etc.)

Machine-executable cross-reference linkage between pictures.

E. The On-Line Calculator

An on-line calculator will be included as part of the redesigned NLS presently being implemented.

Final specifications for this calculator have not yet been determned, but the following subsystems should be available to the user. The on-line calculator will actually be made of two major subsystems -- the NLS Desk Calculator, and the NLS Processor.

1. NLS Desk Calculator

Within the normal NLS command structure, the user will have a small calculator package, giving him the equivalent of a modern programmable electronic desk calculator. The operations to be made available in this subsystem break into two packages: direct operation and semidirect operation.

a. Direct Operation

The user will be able to declare any one of a number of operations (e.g. add, subtract, multiply, divide) and perform simple calculations on NLS word entities. The result will be placed in an accumulator and three other registers, under user control. Normal NLS commands will be used to place results in the appropriate place in a file.

b. Semidirect Operation

The user will also be able to defer specification of the operations to a simple function embedded within the regular NLS text. This function will accept bug selections or keyboard entries as arguments, and return results in an accumulator register on the display. As with direct operation, normal NLS commands will be used to place the

result in the appropriate place in an NLS file.

2. NLS Processor

A system will be implemented to allow the user to call an arbitrarily complex processor (written in an ALGOL-like language) to operate on data stored in a conventional NLS file, and return with text, numerical, or graphic results that may be automatically inserted in the file (under user control).

This will make it possible to generate graphs and charts from numerical data in NLS statements, and to perform more complex calculations than could be performed with semidirect use of the NLS Desk Calculator.

The specifications for the on-line calculator should be completed by mid-March 1969, and the calculator implemented soon after that date.

IV FACILITY DEVELOPMENT AND MAINTENANCE

A. Status

1. Time-Sharing System

We have continued to follow closely the evolution of the Time-Sharing System (TSS) under Project GENIE at Berkeley.

We are currently running the TSS 1.96 system, which includes some improvements in user features over the earlier TSS 1.94.

This version also provides for scratch files on the disc as well as the earlier KDF disc file system.

In the last few months project GENIE has reduced its effort on the time-sharing system, and we will have to pick up any continued evolution.

2. Hardware

Both Tasker display systems have been delivered and accepted.

Both systems are operating reasonably well and are almost up to the expected performance as discussed in Quarterly Report 1 under the present contract (Ref. 1).

The Friden keyboards have been overhauled by Friden, but they are still marginal in performance. The maintenance required to keep them in operation is much too high.

We have ordered two keyboards from Ikor for evaluation. These should be delivered in the next few weeks.

Significant improvements have been made in the video system.

We have improved the monitoring and distribution system by the addition of amplifiers, and we have added equipment for split screen and video mixing. This equipment was extremely valuable for our presentation at the FJCC, and we are now using it in a permanent setup for making movies.

We are currently making movies entirely over the video system. A film service in San Francisco will make sound films directly from a TV monitor. With the mixing and special-effects equipment, combinations of computer-generated views and live views of people and facilities are combined to produce the film.

3. Computer-Generated Sound

Work is progressing at a low level of effort on the implementation of a system for carrying sound signals from the computer to the individual consoles.

These signals will initially be used to carry feedback information on the state of user operations being carried out. Other uses for this information channel may be expected to arise as soon as it is available.

B. Study of System Loads

Considerably more study has been done on the problem of serving NLS through the time-sharing system.

The results of these studies show that we will not be able to serve more than 6 users at a time with reasonable response under the present hardware-software system.

The study also develops a proposed solution to this situation. In particular, we plan some rather extensive rewriting of the NLS software and compilers and the provision of external storage for display buffers. The remainder of this section describes studies in this area. There is a further discussion of system loads in Appendix B.

1. Load Factor

a. Definition

The load factor is the ratio of (number of pages needed to swap in to run a user) to (number of memory pages available for swapping).

Swapping is one of the two critical factors that govern the system's response time to users. The other factor is compute-boundedness.

Swapping is crucial because it takes a large, constant amount of time to bring the pages necessary to answer a user request from the drum to core.

When the requests exceed the available time, there is no remedy: user activity must slow down accordingly.

The empirical data was gathered using a constant, dense system load. Individuals sat at consoles repeating a single command to recreate the display at a constant rate. New

users were added one by one. During the entire test, a program measured and recorded the average maximum time that forks were spending the in long-quantum overflow queue. The load factor was defined in such a way as to match these results.

b. Assumed System Load

All of this analysis attempts to assume that the system is operating under a constant, favorable load. We count on the Monitor to be resident and to occupy seven pages. NLS shared code is also assumed to be in memory and to occupy ten pages. Finally, one page is reserved for the disc operation.

This means that 18 of the 32 core pages are taken before writable code is assigned for the first user.

This is, of course, an artificial assumption. Anything else that happens changes the number of available pages.

The files used during the testing were very short ones, so that no file reads or writes were being done. This eliminates a dimension of complexity from the analysis, but it makes the results even more artificial.

File drum requests have an adverse multiplicative effect on the swapping load because there is a reasonable chance that nothing useful can be accomplished during the read. If a user program is waiting for a read and another must be activated, a swap may have to take place; but then it may be necessary to swap back pages to reactivitate the user waiting for the request, thus doubling the swaps.

We feel that the problem of drum requests should be solved by using better heuristics in file organization and thus maximizing the use of each request.

c. Multiplier Factor

The hardest question for prediction or measurement concerns the average load that a number of users put on the system during their normal work.

The averages may not be taken over a long period of time. The problem is determining the probability of multiple user requests within a short, constant time increment -- for example, the probability that 3 users will request a display creation within 1/2 second of each other.

We cannot even guess at these probabilities. However, we feel that the maximum load factor tolerable to users working at normal rates is three times the maximum tolerable load factor for users working at the steady, dense request rate of the tests. This is 1.2*3 or 3.6.

2. Display-Creation Swapping-Load Factor

a. Peak-Load Operation

The analysis is concerned with the system load when all users are creating new display images.

The predictions assume that the display-creation package does not become compute-bound. The current version does become compute-bound, and consequently causes considerably more swapping.

b. Derivation of Figures

Nonshared: Each new user requires 3 writable pages of core, besides the page that is already frozen for him.

Thus the numerator is 3*n.

The denominator is 14-n, because a new page is frozen as each user is added.

Shared: Each new user requires 4 writable pages. The fourth page is required because the display creation needs the table space. These tables currently reside in the page with the frozen display buffer. If the buffers were kept by the Monitor, the extra page would still be needed.

The numerator is 4*n.

The denominator is 14-IP((n+1)/2), assuming that each user will require 1/2 page of display buffer. Since the pages are frozen in whole increments, the denominator decreases when odd-numbered users are added.

External: Each new user requires 4 new pages of writable memory, just as in the shared case. However, since no memory is ever frozen, the denominator is constant.

The numerator is 4*n.

The denominator is 14.

Figure 1 is a graph of the data for these three cases. Table I shows the numerical data. (Note: Illustrations and tables are at the end of Section IV-B.)

3. Comparison to Data

a. Experiments

Three of the tests performed are relevant to the determination of the validity of the load factor.

The data collected was the average time (in tenths of milliseconds) that the measuring program spent in the long-quantum overflow queue. This is an indication of the maximum time any program waited for activitation.

Thus we are assured that response to NLS users was always shorter than the waiting time for the measuring program.

The first test had the users typing "J I ca cd" (ca is command accept, cd is command delete) at about one character per second. This has the effect of invoking all the input-feedback machinery, but not the text-editing or display-creation code. The information thus shows the load put on the system during command selection, entity selection, and literal typing.

The times were 46, 49, 54, 61, 77, and 107 for 1 to 6 users respectively.

Figure 2 is a graph of these input-feedback response times.

In the second test each user typed "J I ca ca" at about the same rate. The Display image was constantly being recreated after the second ca. The display image had 10 statements, each only 3 characters long. Thus, in a certain sense, the image being created was minimal.

The times were 50, 55, 65, 110, and 29,500 for 1 to 5 users respectively. When the response time was more than 3 seconds, users slowed down to wait for the system to respond and all the measurements became meaningless.

Figure 3 is a graph of these short-line display-creation response times.

The third test was the same as the second except that each statement was a full line long (about 70 characters).

The times were 53, 71, 111, 4000, and 76,000 for 1 to 5 users respectively. Again the 0.4-second response time began to bother users, and the 7-second time destroyed the meaning of the expreiment.

Figure 4 is a graph of these long-line display-creation response times.

b. Maximum Load Factor

During the input-feedback test only 1 page of swappable core is required for each user. Thus the load-factor ratios are 1/13, 2/12, 3/11, 4/10, 5/9, 6/8, 7/7, and 8/6 for 1 to 8 users. The response-time curve rises so fast from 5 to 6 users that we expect it to be out of range at 7 or 8. If we optimistically assume 8, the maximum tolerable load factor during high density use is about 8/6 or 1.3.

During the display-creation test, 3 swappable pages are required for each user. Thus the load-factor ratios are 3/13, 6/12, 9/11, 12/10, and 15/9 for 1 to 5 users. In this case 4 users had only a 0.011-second response delay, while 5 users produced a totally unacceptable load. This again indicates that the maximum load factor was about 12/10 or 1.2.

The third test clearly demonstrates the result of compute-boundedness. The number of users getting tolerable response was again 4, but the response was barely tolerable. At 5 users the system was compute-bound. At that point all else is irrelevant. More service is requested than can possibly be given. Thus, for full display images the time necessary to recompute the entire display image is also a limiting factor.

Again we must stress that we have not been able to devise any way of measuring the average load under normal use or the probability of simultaneous service request by users. The multiplying factor of 3 is our intuitive guess: it is just a guess and may be completely wrong.

4. Conclusions

The most surprising fact revealed by the study is the similarity of shared and nonshared load factors.

We originally thought that sharing of display buffer space would result in a significant saving and allow us to run 12 users.

This study shows, however, that there is nothing to be gained by sharing buffer space.

The study also shows that it is necessary to make more memory available for swapping if more than 7 users are to receive adequate service.

The only feasible way to do this is with an external memory, making more core available by not freezing any pages.

It is also clear from the external curve that more that 12 users would be an unreachable goal. The load factor for 12 is 3.14, right at the limit.

The third shows that to reach the maximum of twelve users the system must never be compute-bound. If it goes compute-bound at, say, 8 users, even external memory will not help. The response time immediately increases to seconds until the load is serviced.

Our best estimates indicate that the display image could, on the average, be created in a short quantum (32 milliseconds).

If this could be done the only other thing to slow down response would be the random-file drum reads.

This study has purposely avoided mention of queues, other subsystems, and their effects on the system. These problems deserve another study. In order to increase the service to 12 users, all the limiting factors must be overcome. It would do no good at all to get external memory without solving the compute-boundedness problem or vice versa. Neither would it do any good to add or change queues without also attacking the problem of swappable memory and long-quantum overflow.

5. Study of NLS Usage

a. Method

For each NLS user, the console number and the "log" of the number of clock cycles (actually the smallest integer greater than the log) between events for that console were recorded .

The "log" was written using the ith letter of the alphabet to stand for i.

Thus '5 g' meant console 5 had taken between 2↑f and 2↑g

cycles, where f=6 and g=7.

Table II gives the "logs" and their corresponding times.

Figure 5 is a graph showing the distribution of the intervals between display-creation actions from individual users, and Table III gives the numerical data.

Figure 6 is a graph showing the distribution of the intervals between character-input actions from individual users, and Table IV gives the numerical data.

Fig. 1 LOAD FACTOR AS A FUNCTION OF NUMBER OF USERS

Illustration produced by means of NLS Vector Package and Film Output (see Sections III-D-1 and V-A-1-d)

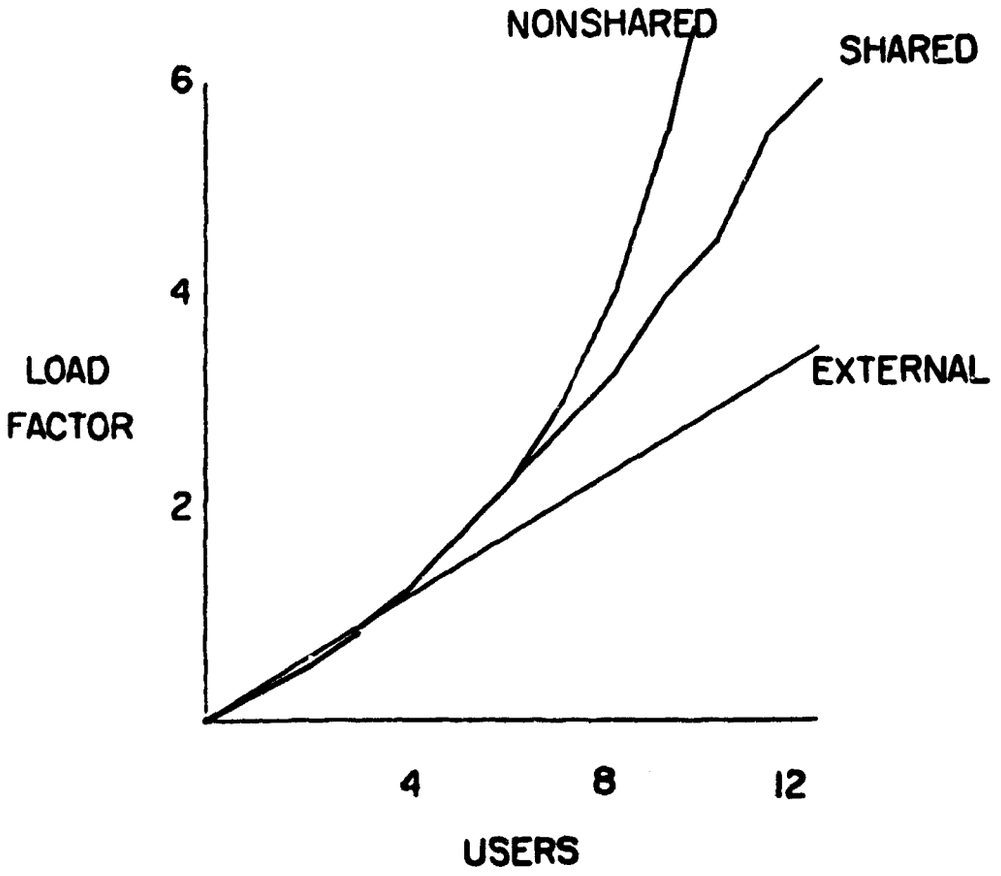

Fig. 2 INPUT-FEEDBACK RESPONSE TIMES (in tenths of milliseconds)

Illustration produced by means of NLS Vector Package and Film Output (see Sections III-D-1 and V-A-1-d)

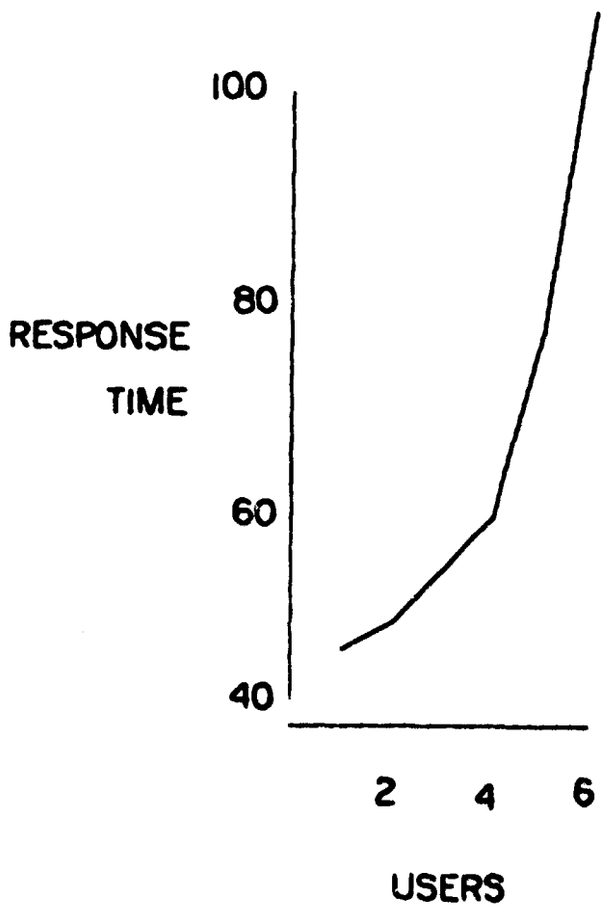

Fig. 3 SHORT-LINE DISPLAY-CREATION RESPONSE TIMES (in tenths of milliseconds)

Illustration produced by means of NLS Vector Package and Film Output (see Sections III-D-1 and V-A-1-d)

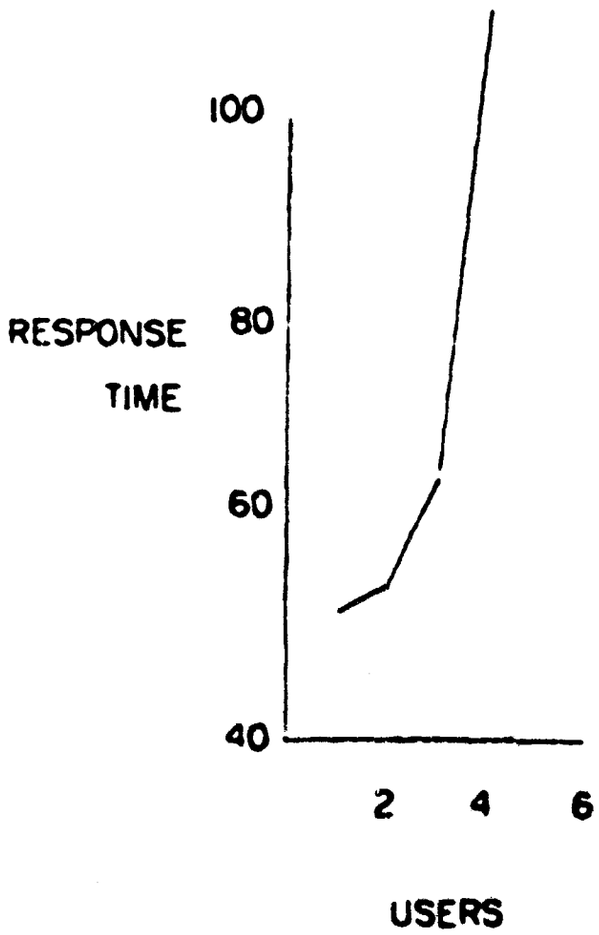

Fig. 4 LONG-LINE DISPLAY-CREATION RESPONSE TIMES (in tenths of milliseconds)

Illustration produced by means of NLS Vector Package and Film Output (see Sections III-D-1 and V-A-1-d)

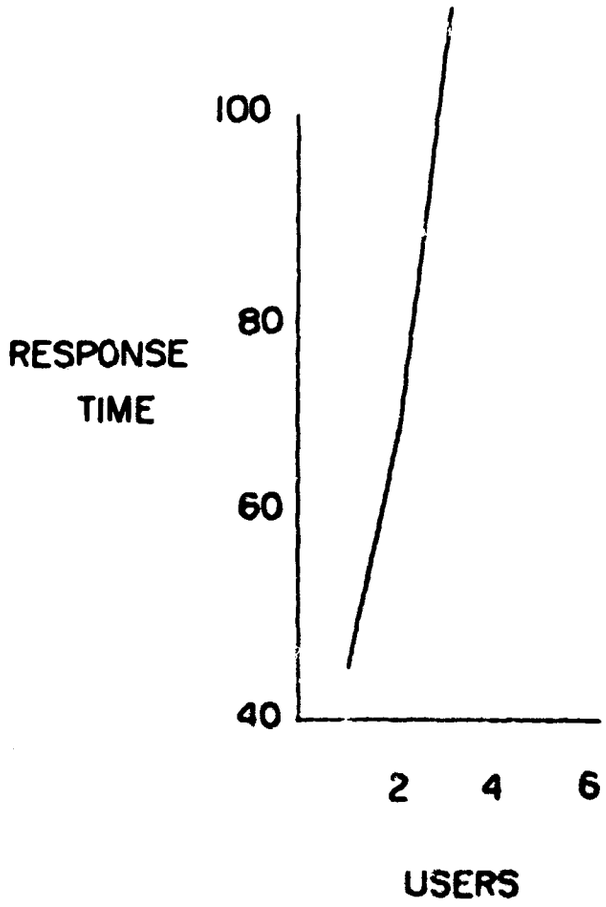

Fig. 5 DISTRIBUTION OF INTERVALS BETWEEN DISPLAY-CREATION ACTIONS BY INDIVIDUALS

Illustration produced by means of NLS Vector Package and Film Output (see Sections III-D-1 and V-A-1-d)

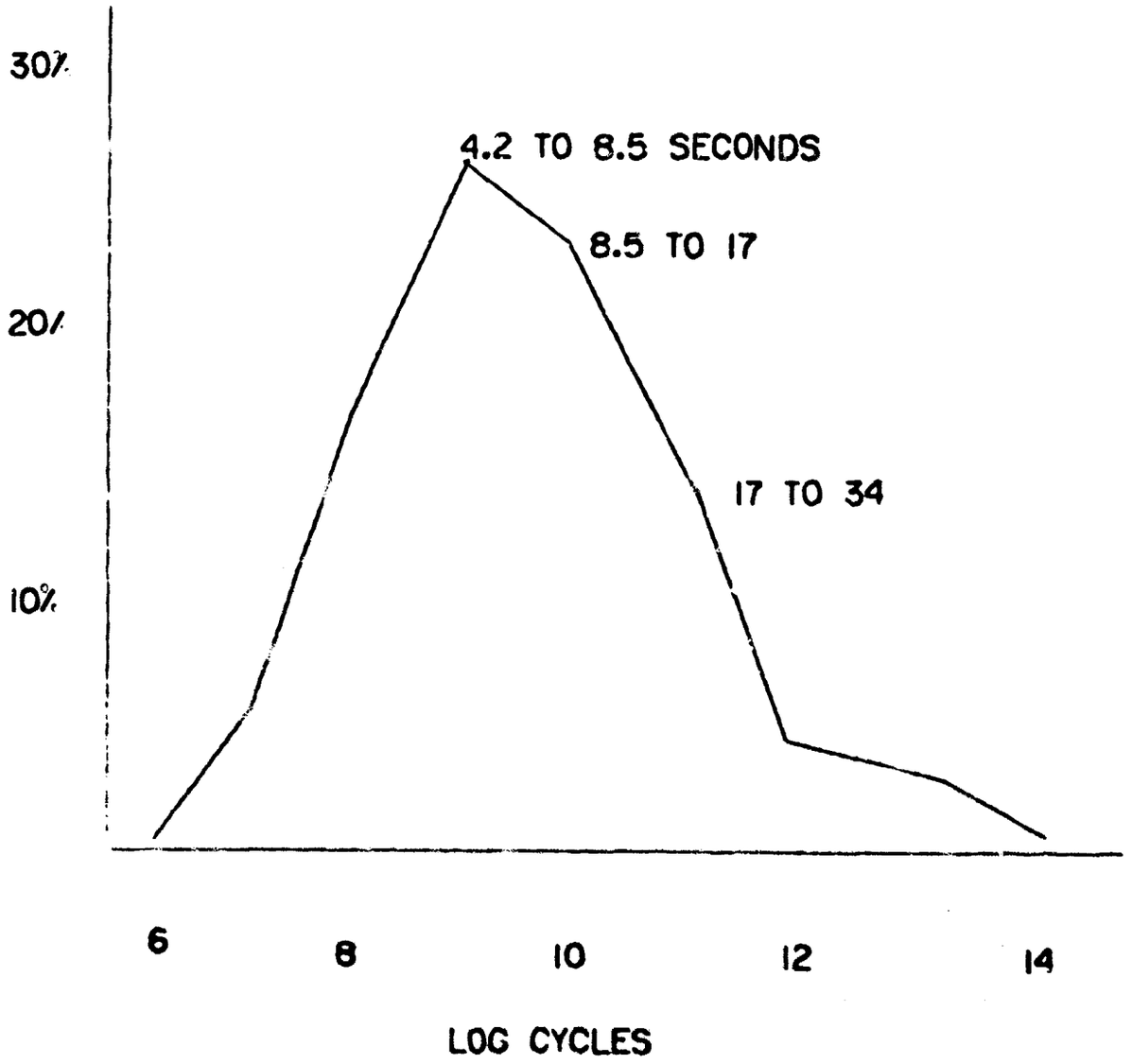

Fig. 6 DISTRIBUTION OF INTERVALS BETWEEN CHARACTER-INPUT ACTIONS BY INDIVIDUALS

Illustration produced by means of NLS Vector Package and Film Output (see Sections III-D-1 and V-A-1-d)

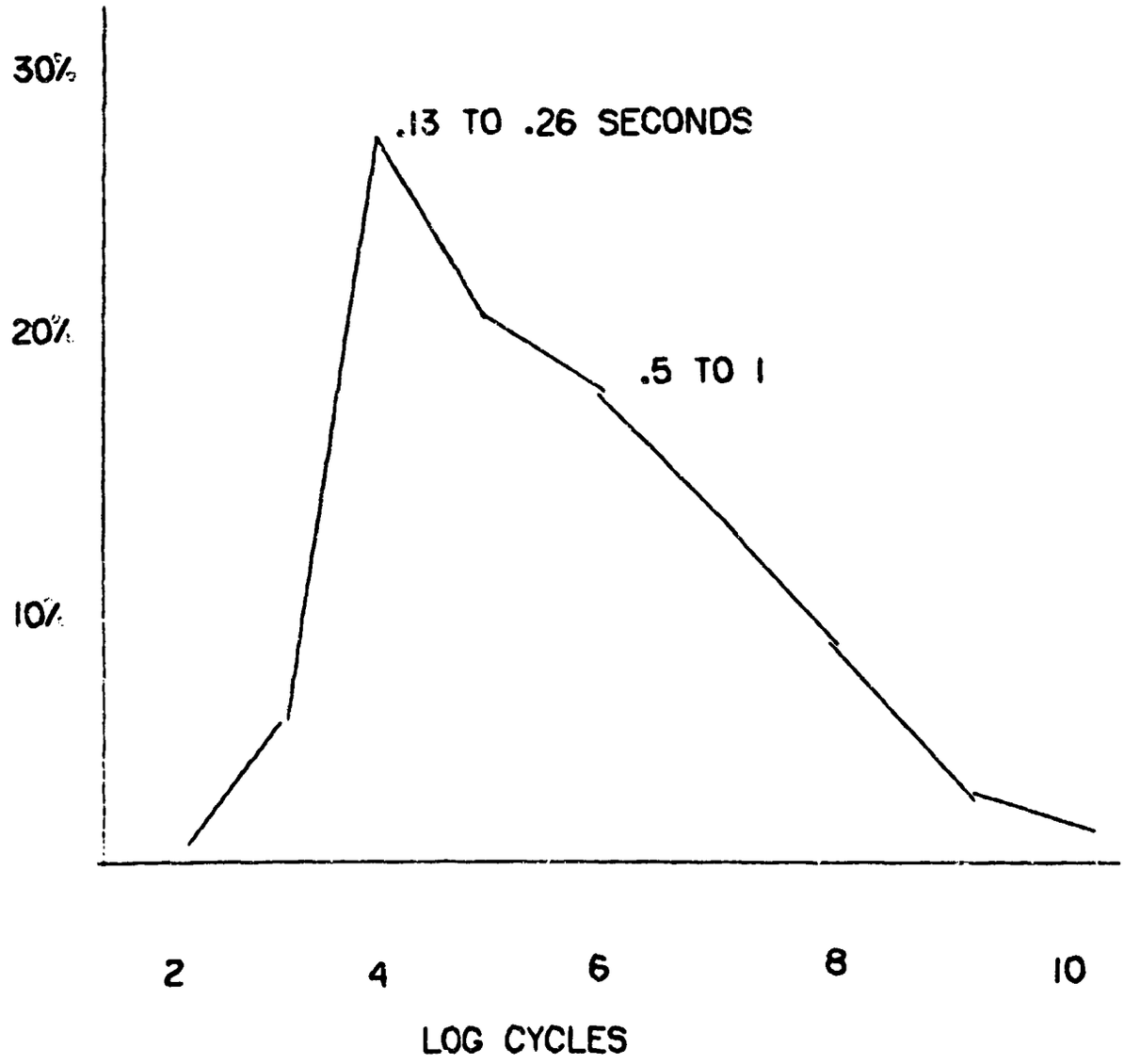

Table I: LOAD FACTOR AS A FUNCTION OF NUMBER OF USERS (see Fig. 1)

Users	Nonshared	Shared	External
0	0/14	0/14	0/14
1	3/13	4/13	4/14
2	6/12	8/13	8/14
3	9/11	12/12	12/14
4	12/10	16/12	16/14
5	15/9	20/11	20/14
6	18/8	24/11	24/14
7	21/7	28/10	28/14
8		32/10	32/14
9			36/14
10			40/14
11			44/14
12			48/14

Table II: LOGS AND CORRESPONDING TIMES (clock runs at 60 Hz)

Log	Time			
c-3	0.06	to	0.13	sec
d-4	0.13	to	0.26	sec
e-5	0.26	to	0.53	sec
f-6	0.53	to	1.06	sec
g-7	1.1	to	2.1	sec
h-8	2.1	to	4.3	sec
i-9	4.3	to	8.5	sec
j-10	8.5	to	17.1	sec
k-11	17.1	to	34.1	sec
l-12	34.1	to	68.2	sec
m-13	1.13	to	2.27	min
n-14	2.27	to	4.55	min
o-15	4.55	to	9.10	min
p-16	9.10	to	18.20	min

Table III: INTERVAL DISTRIBUTION OF DISPLAY CREATION BY INDIVIDUALS (see Fig. 5; number of readings = 1214)

Log	Number	Percentage
2	0	0.0
3	2	0.2
4	1	0.1
5	4	0.3
6	19	1.6
7	82	6.8
8	190	15.6
9	308	25.4
10	269	22.2
11	179	14.7
12	81	6.7
13	53	4.4
14	17	1.4
15	6	0.5
16	2	0.2
17	0	0.0

Table IV: INTERVAL DISTRIBUTION OF CHARACTER INPUT BY INDIVIDUALS (see Fig. 6; number of readings = 1593)

Log	Number	Percentage
1	65*	4.1
2	12	0.7
3	91	5.7
4	399	25.0
5	312	19.6
6	282	17.7
7	206	12.9
8	141	8.9
9	53	3.3
10	13	0.8
11	12	0.7
12	5	0.3
13	1	

*NOTE: The reading for 1 is caused by people typing ahead of the system.

C. Reliability of Data

1. Bryant Disc-File System

The 32-megaword Bryant disc file is in regular operation both for permanent storage of files and as user scratch memory space.

Permanent storage is under a software system called KDF, developed by Project GENIE. This has been in use for several months and has proved highly reliable. Disc storage is far more convenient to the user than tape storage, because of the much faster access and elimination of the need to mount tapes.

The disc has been used for user scratch memory for only a few weeks. In this capacity it replaces the drum, thus freeing a great deal of drum space for swapping and reducing the system load.

Reliability was initially poor, but is now quite good. Files on disc scratch space may be made permanent, and it is expected that disc scratch files will be used for permanent storage in the near future. The advantage of this is that KDF can access only a very limited amount of disc space for each user, whereas scratch space is limited only by the very large size of the disc itself.

2. File Cleanup

Every NLS file contains not only the actual text written into it but also a great deal of information used by the system in transferring it from one location to another, relating statements to one another structurally, displaying the file, etc.

Occasionally, hardware errors cause bad spots in this data or bad characters in the text; also, because of the complexity of the software and its constant state of revision, software bugs sometimes cause errors in file information.

Such errors can be very serious in their effect on a file. Certain types make the file completely unreadable by NLS, and these must be corrected by accessing the file directly in core with an on-line debugging system. Many errors, however, affect the user's use of the file only when a particular statement is displayed or a particular operation attempted.

A user command in NLS executes a program that can automatically correct many of these errors and can detect and list others.

This system, called File Cleanup, has drastically reduced the number of files lost through system errors, particularly those errors that are not immediately apparent but that may propagate into other, more serious errors when some operation is executed on the file.

V GENERAL DEVELOPMENTS OF RELEVANCE

A. Development of Hard-Copy Output

The development of systems to transfer displayed information to hard copy has been in progress since the implementation of the CDC 3100 On-Line Text System (NLTS). Current work in this area is only partly supported under this contract. The principal subsystem concerned with generation of hard copy is the "PASS4" output processor, described below.

1. The "PASS4" Output Processor

a. Introduction

PASS4 is the name of a program that takes an NLS file as input and produces a formatted output file for some external device (paper-tape-driven automatic typewriter, line printer, Teletype, or CRT-to-microfilm process).

Input is performed one statement at a time.

Text is formatted one line at a time for output according to established conventions. However, much of this process is under control of "directives" which enable the user to change various parameters governing the output process.

The file is translated and sent to a user-designated output file in the character code of a designated external device.

The resulting output file may then be transferred to the desired output medium (such as paper tape, hard copy, magnetic tape, film, etc.).

b. Directives

A "directive" is a machine-recognizable string of text embedded in a document, that "directs" the output processing and formatting being performed on the document.

Directives provide a means of changing various parameters, code conversion tables, etc. that are used in the actual output formatting process. Since these directives are embedded as text in a document, they can be edited and manipulated as text, and yet be recognized on output as having special meaning.

When directives are recognized in the output process, the parameters are changed or some action is performed at the point in the output process where the directive occurs.

Thus, directives need to be placed at the point in the text where the particular parameter change (or action) is desired.

On output it is possible to supress the printing of directives so that the output will be affected, but the output will not include (as text) the actual directives.

c. Features of Hard-Copy Formatting

The directives provide a great deal of user control over a large number of special format features. The following is a list of some features that are commonly used in hard-copy production. There are also numerous other features used more rarely for special effects.

Automatic pagination, with automatic numbering of pages in either Arabic or Roman numerals, starting from any given number

Automatic formatting of lines, with or without insertion of spaces to achieve a justified right margin

Output of some preset "running header" at the top of each page

Suppression of statement numbers

Substitution of some preset number of spaces for each statement number

Suppression of indentation of statements according to level

Automatic page restore at the beginning of each new major section of the file (i.e., pagination before each first-level statement)

Centering of specified lines on the page.

d. Output Devices

Automatic Typewriters: Paper tape can be punched to drive either a Flexowriter or a Dura Mach 10 typewriter for production of high-quality final typing.

Drawings created in NLS as part of a file are lost on output to a typewriter.

Line Printer: A file may be set up to serve as input to a Potter line printer with a full character set including upper- and lower-case characters.

Hard copy produced on the printer is identical to Dura typewriter output, except that the type is of lower quality and drawings are approximated by plotting with available characters. The quality of these "drawings" is usually good enough for intelligibility and for examination of page format.

Printer output is used where rapid production is desired, as for drafts and internal memoranda.

Film Output: Output via a CRT-to-film process permits high-quality hard-copy reproduction of mixed text and drawings created in NLS. This process is currently under experimental development but is expected to become a primary means of hard-copy production.

Teletype Output: Teletypes are not ordinarily used for hard-copy production. A teletype-formatted output file is useful as an intermediate stage in certain file-conversion processes, since it may be read by NLS as if the text were being typed in from a console.

B. AHIRC Presentation at the FJCC

On 9 December 1968, the AHIRC conducted a major technical session at the Fall Joint Computer Conference in San Francisco. Participation in this conference was not sponsored by RADC. Because the development of some of the material presented by the Center at this conference has been supported by RADC, and because of the positive impact of the presentation on all aspects of the Center's growth and activity, a full description of the FJCC presentation is given in Appendix A.

C. Expansion of the AHI Research Center

Towards the end of the period covered by this report, the AHI Research Center issued three proposals to ARPA for expansion of various aspects of our program. Some of these aspects are supported under the present RADC contract. These proposals are described in detail in Appendix B to this report.

GLOSSARY

AHIRC -- Acronym for the Augmented Human Intellect Research Center at Stanford Research Institute.

AHI -- Acronym for the concept "augmented human intellect".

ARPA -- Acronym for the Advanced Research Projects Agency.

Augmentation -- Used in this report to indicate the extension of human intellectual and organizational capabilities by means of close interaction with computer aids and by use of special procedural and organizational techniques designed to support and exploit this interaction.

Bootstrapping -- A name for the research strategy of the AHIRC. Simply stated, it means that the immediate aim of any effort within the AHIRC is to increase the effectiveness of the AHIRC. Thus the AHIRC develops tools for making better tools, etc.

Center -- Another term used for the AHIRC.

Console -- As used here, this means specifically a user's control console for the AHIRC's On-Line System (NLS). The consoles presently in use consist of a display screen, a keyboard, a "mouse", and a "keyset".

File -- As used here, this refers to a unified collection of information held in computer storage for use with the On-Line System. A file may contain text (English or program code), numerical information, graphics, or any combination of these. Conceptually, a file corresponds roughly to a hard-copy document.

FJCC -- The 1968 Fall Joint Computer Conference in San Francisco. The AHIRC conducted a major presentation at the FJCC.

GENIE -- Project GENIE, at the University of California at Berkeley, developed the time-sharing software for the SDS 940 computer used by the AHIRC.

Keyset -- A device consisting of five keys to be struck with the left hand in operating the On-Line System.

Mouse -- A device operated by the right hand in using the On-Line System. The mouse rolls freely on a flat surface, causing a cursor spot on the display screen to move correspondingly.

Network -- The planned Advanced Research Projects Agency network of research computer installations.

GLOSSARY

NIC -- The Network Information Center, to be incorporated in the ARPA network. The NIC will operate as a computer-assisted library service for information pertaining to the network, to be used by network members, and will be operated by the AHIRC.

NLS -- Acronym for the On-Line System.

On-Line System -- This is the AHIRC's principal and central development in the area of computer aids to the human intellect. As presently constituted, it is a time-shared multiconsole system for the composition, study, and modification of files (see definition of "file"). The system is described in detail in the body of this report.

PASS4 -- An output-processing program used to convert NLS files to hard-copy format for output via one of a number of different devices.

RADC -- Acronym for Rome Air Development Center.

SRI -- Acronym for Stanford Research Institute

Statement -- The basic structural unit of a file. A statement consists of an arbitrary string of text, plus graphic information. A file consists of a number of statements in an explicit hierarchical structure.

TSS -- Acronym for Time-Sharing System. Specifically, the system developed by Project GENIE for the SDS 940 computer.

REFERENCES

1. D. C. Engelbart, W. K. English, and D. A. Evans, "Study for the Development of Computer-Augmented Management Techniques," Quarterly Progress Report 1, Contract F30602-68-C-0286, SRI Project 7101, Stanford Research Institute, Menlo Park, California 94025 (October 1968).

2. D. C. Engelbart, "Augmenting Human Intellect: A Conceptual Framework," Summary Report, Contract AF 49(638)-1024, SRI Project 3578, Stanford Research Institute, Menlo Park, California 94025 (October 1962), AD289565.

3. D. C. Engelbart, W. K. English, and J. F. Rulifson, "Development of a Multidisplay, Time-Shared Computer Facility and Computer-Augmented Management-System Research," Final Report, Contract AF 30(602)-4103, SRI Project 5919, Stanford Research Institute, Menlo Park, California 94025 (April 1968).

4. D. C. Engelbart and Staff of the Augmented Human Intellect Research Center, "Study for the Development of Human Intellect Augmentation Techniques," Semiannual Technical Letter Report 1, Contract NAS1-7897, SRI Project 7079, Stanford Research Institute, Menlo Park, California 94025 (February 1969).

BIBLIOGRAPHY

The following is a list of publications from the AHI Research Center which are not cited directly in this report.

D. C. Engelbart, "Special Considerations of the Individual As a User, Generator, and Retriever of Information," Paper presented at Annual Meeting of American Documentation Institute, Berkeley, California (23-27 October 1960).

D. C. Engelbart, "A Conceptual Framework for the Augmentation of Man's Intellect," in Vistas in Information Handling, Volume 1, D. W. Howerton and D. C. Weeks, eds., Spartan Books, Washington, D.C. (1963).

D. C. Engelbart, "Augmenting Human Intellect: Experiments, Concepts, and Possibilities," Summary Report, Contract AF 49(638)-1024, SRI Project 3578, Stanford Research Institute, Menlo Park, California (March 1965), AD640989.

D. C. Engelbart and B. Huddart, "Research on Computer-Augmented Information Management," Technical Report ESD-TDR-65-168, Contract AF 19(628)-4088, Stanford Research Institute, Menlo Park, California (March 1965), AD622520.

W. K. English, D. C. Engelbart, and B. Huddart, "Computer-Aided Display Control," Final Report, Contract NAS1-3988, SRI Project 5061, Stanford Research Institute, Menlo Park, California (July 1965).

W. K. English, D. C. Engelbart, and M. L. Berman, "Display-Selection Techniques for Text Manipulation," IEEE Trans. on Human Factors in Electronics, Vol. HFE-8, No. 1, pp. 5-15 (March 1967).

D. C. Engelbart, W. K. English, and J. F. Rulifson, "Study For The Development of Human Intellect Augmentation Techniques," Interim Progress Report, Contract NAS1-5904, SRI Project 5890, Stanford Research Institute, Menlo Park, California (March 1967)

J. D. Hopper and L. P. Deutsch, "COPE: An Assembler and On-Line-CRT Debugging System for the CDC 3100," Technical Report 1, Contract NAS 1-5904, SRI Project 5890, Stanford Research Institute, Menlo Park, California (March 1968).

R. E. Hay and J. F. Rulifson, "MOL940: A Machine-Oriented ALGOL-Like Language for the SDS 940," Technical Report 2, Contract NAS 1-5904, SRI Project 5890, Stanford Research Institute, Menlo Park, California (April 1968).

(continued overleaf)

BIBLIOGRAPHY

D. C. Engelbart, "Human Intellect Augmentation Techniques," Final Report, Contract NAS 1-5904, SRI Project 5890, Stanford Research Institute, Menlo Park, California (July 1968).

Note: Reports with AD numbers are available from Defense Documentation Center, Building 5, Cameron Station, Alexandria, Virginia 22314.

Appendix A
AHI PARTICIPATION IN THE FJCC

1. Preface

The work described in this section was not funded by RADC; it is included in this report for purposes of information and because some of the developments reported will be relevant to work undertaken for RADC.

2. Introduction

On 9 December 1968, D. C. Engelbart made a presentation to the 1968 Fall Joint Computer Conference in San Francisco. This constituted an entire session, of which Dr. Engelbart was chairman.

The presentation was a demonstration of the special techniques and capabilities developed by the Center; interactive computer manipulation of text, with real-time CRT display, was used as the medium for describing, demonstrating, and discussing the Center's work in developing the capabilities that were being demonstrated.

The user console, the projection equipment, and the video control equipment were located in the lecture hall; all other equipment -- the computer itself, the CRT display equipment, etc. -- remained at Stanford Research Institute in Menlo Park. Video, audio, and control information were transmitted via a microwave link leased from the telephone company.

This "computer medium" was closely coordinated with the use of speech and of advanced video techniques, and the resulting combination was communicated to the audience by means of projection television, employing an Eidophor projector borrowed from NASA-Langley Research Center.

The television image was used to carry pictures of the computer CRT display, the faces of speakers, and equipment in the AHIRC computer room at Stanford Research Institute.

During the remaining two days of the conference, the AHI Research Center held open house in a specially prepared room at the conference.

The reaction of conference participants was highly enthusiastic, and greatly increased public and professional interest in the Center's activities is expected as a consequence.

The Center's participation in the FJCC was also a testing program for a number of new and projected developments in the Center's

work. The Conference program was the culmination of a month of intensive preparatory work.

3. Preparation

The preparatory work fell into three major categories: new hardware and modification of existing hardware, new software, and the materials, techniques, and scenario for the actual presentation.

a. Hardware Preparation

The principal effort in the hardware area was concerned with the generation, processing, mixing, transmission, and projection of visual images.

Both SRI and Tasker personnel did considerable work on the primary displays and display-generation equipment so as to have them in peak operating condition.

Considerable work was also done on television systems, for use in mixing various images in various ways.

Considerable effort went into familiarization with the Eidophor projector and adjustment of its interface with the SRI equipment.

The console used for the presentation was of a new design by Herman Miller Research.

The old-style consoles used by AHIRC consist of a table, with the display monitor partially recessed in the top and the three control devices (keyboard, keyset, and mouse) at the front edge.

The new console has the monitor mounted by itself on a movable stand (wall-mounted or free-standing) and the controls mounted on a tray attached to the user's chair.

b. Software Preparation

The software preparation included programming of several new features.

Provision was made to link two separate time-shared consoles for collaborative work, by displaying the cursor spots for both consoles on each display screen. The same information is displayed on each console (by video switching, under

manual control), with one user in control of the system and the other communicating with him via an audio link. The second user uses his cursor as a pointer when making reference to information on the display.

New commands were implemented to give the user direct control over the formatting of the display. This control permits detailed restructuring of the space allocations on the screen, with the primary purpose of permitting special formatting for use with video linking. Thus the user may format his screen so that all computer-generated display appears only on the left half of the screen, leaving the right half free for video of another user's face, or another user's display, etc.

The software for the first stage of a computer-generated sound system was implemented. This system will provide each console with a sound signal modulated in various ways to carry real-time information on the internal processes of the system, as relevant to the individual user.

As implemented for the FJCC, this system provided sound to only one console and carried only limited information.

c. Presentation Preparation

The basic plan for the presentation was to use text information held in computer files as the foundation for the development of the entire presentation.

Accordingly, a complex structure of relevant files, both new and existing, was established; cross-references were inserted and a scenario was developed in terms of topics and the files that would be used for developing each topic.

4. Main Presentation

No attempt is made here to give details on the content of the presentation. Interested persons are referred to the paper published in the Proceedings of the 1968 Fall Joint Computer Conference. A kinescope film was made of the presentation.

a. General Description

The main presentation took place in the Arena of the San Francisco Civic Auditorium. Instead of using a podium, Engelbart sat at one side of the stage, with a console. A forty-foot screen was hung at the center of the stage, and the

video picture was projected on this screen by the Eidophor projector. A control center was set up at the back of the Arena, along with the projector.

Engelbart wore a headset with boom microphone; sound picked up by the microphone was amplified into the Arena, and the headset carried communications from the control crew.

The computer-generated sound was also amplified into the Arena during portions of the presentation.

Two sections of the presentation were made from remote consoles in Menlo Park.

The first of these, given by J. F. Rulifson, was concerned with special software methods used in AHIRC.

Special techniques used for this section included superimposing the speaker's face upon a view of his display for general discussion purposes, and blanking out the face for detailed discussion of the displayed material.

The second remote section, by W. H. Paxton, covered information-retrieval techniques.

The beginning of this section included a demonstration of the linking capability; the projection screen carried Paxton's display with both his own tracking spot and Engelbart's, with Paxton's face shown in one corner of the screen (from which the computer display had been cleared by reformatting the display under user control).

An audience of perhaps 800 persons attended the main presentation.

5. Open House

The two-day open-house program, held in two rooms at the Civic Auditorium, consisted of informal demonstrations of AHIRC's On-Line System, informal discussions between attendees and AHIRC personnel, and display of the new Herman Miller Research equipment, set up as a complete office.

Approximately 1500 persons signed a register; the estimated total attendance at the open house was 2000.

6. Results

At present it is only possible to estimate some of the results of the FJCC program. The benefits fall into two categories: improved communication and relationship with the professional computer community ("external" benefits) and benefits directly affecting the internal workings and research of the AHI Research Center ("internal" benefits).

a. External Benefits

The number of professional people who are acquainted with AHI research has been vastly increased. The professional press has shown considerable interest in AHI research as a result of the presentation. Considerable local newspaper coverage also resulted.

At a luncheon for conference participants, special official recognition was given to the session presentation.

b. Internal Benefits

The most immediate internal benefit has been the opportunity for a general shakedown of working procedures and a considerable stimulation of new ideas for future development. Very valuable experience was also gained in several categories.

Experience in Hardware Techniques: A great deal of valuable experience was gained in the coordination and use of television equipment, not only for presentation purposes but as a technique for use in the AHI program itself. Use of such techniques as aids to interpersonal collaboration in the use of interactive display consoles is expected to become an important area of AHI research.

Experience in File Usage: The first heavy and extensive use of linked-file structures occurred in connection with the conference, and much useful information was gained.

Experience in Collaborative Techniques: Some collaborative-working techniques were tried for the first time in connection with the conference, and the experience will be of very great value since this area is one of the most important areas of planned AHI work for the future.

Appendix A
AHI PARTICIPATION IN THE FJCC

Experience in Presen tation Techniques: The session presentation was by far the largest and most elaborate presentation ever mounted by AHIRC. The experience of using interactive computer techniques as a medium for communication will be of the greatest value.

Appendix B
PROPOSALS FOR EXPANSION

During this last period we issued three proposals to ARPA for expansion of various aspects of our program.

We have developed specific plans for the next few months, based upon the acceptance of Proposal I. If the others are accepted, we will modify our plans accordingly.

1. Proposal I : "Study for the Development of Computer-Augmented Management Techniques -- a Proposal for Increase in Scope", SRI Proposal No. ESU 68-140, submitted to RADC on 25 November 1968. This proposal has been accepted.

In this proposal we recommend specific hardware developments that will greatly improve the responsiveness of the system for display users.

We propose improvements in "service" software that will allow more efficient use of the facility and much more flexibility in the development of software central to the AHI research goals.

In addition, we propose the development of audio communication facilities and experiments with video communications that will lead to more effective use of the terminals and pursuit of the basic AHI objectives.

2. Proposal II : Interactive Display Subsystem

This is a proposal for increasing the console-support capacity of our current SDS 940 system from its current 6 CRT consoles to 30 and for adding 12 more CRT consoles for local use. This would leave reserve capacity for about 5 NLS users from the network at large, or the equivalent in general network service.

By next year we will need the capacity to handle about 24 interactive graphics terminals (for AHI and NIC staff), plus the equivalent of 5 or 6 such users as remote network service.

We have recently proposed (in a current proposal to RADC) some system modifications that would [illegible] ty up to 12 CRT users plus one or two network-usage equivalents.

The modifications to our system discussed here would be an alternative to the above proposal.

The general basis for this expansion design stems from a dichotomy of service responses. (See Section IV-B for a detailed report of the study of system loads.)

The current saturation of our 940 at about 6 to 7 users of NLS

(the On-Line System) is due to excessive demands on the swapper to handle the many simple feedback responses; the processor itself is definitely not compute-bound.

Upon analysis of the response services provided by the system, there emerges a clear dichotomy that leads to some important new system-organization possibilities:

Class I service responses, requiring access to the file data or otherwise requiring the full time-sharing capability of the 940

Class II service responses, not requiring file access, but rather operating upon a relatively small amount of data (i.e., upon the "context" record for the user's current control state and upon his display-buffer information).

This dichotomy is probably rather typical of interactive systems.

The Class II responses account for the largest number of service transactions, and their high peak rates of occurrence produce the current high, disabling ratio of swapping time to compute time.

These responses entail known, short computations, and can be serviced by simple queuing and executing to completion. In other words, they do not require the whole time-sharing machinery.

We propose to develop a special subsystem, interposed between the 940 and the display stations, that can handle work-station I/O and the Class II responses, leaving the 940 to service only the Class I responses.

It has been previously recommended that this division of servicing should obtain when we accommodate remote network users with NLS -- i.e., the host computer of the remote user should keep the control-state tables, the literal buffers, etc., and should provide the Class II responses.

Much of the Class II service is affected by the particular display devices, etc., and would need to be special to that host in any case.

At least there will have to be a conversion process in that host in order to map input devices to a standard input for us and to map a standard output description

into their output-device drivers.

This plan has been put forward in a memorandum by J. F. Rulifson (SRI/AHI) and S. Carr (Utah).

Thus, this new design would offer an appealing uniformity with respect to the network, in that the 940 would service remote users via the network in exactly the same way as it would service local users via the interface subsystem.

The Class II response programs would be written in our machine-independent Special-Purpose Languages (SPLs), for which (in the Rulifson/Carr proposal) translators would be developed to compile the control programs for either our interactive-display subsystem or the remote host computer.

Important to our approach are the interval-distribution characteristics observed with our NLS users. For Class-I responses, the mean is about 8 seconds,

With this distribution, it is estimated that the 940 could service at least 30 users with response equivalent to what it now provides when it is servicing 4 or 5 users.

In that this same type of interface subsystem could be used with less responsive time-sharing systems to allow them to accommodate some highly interactive users, the need for developing principles of design is rather general.

Also, for future such applications (both ours and others) it seems quite important that the implementation of such an "interface" subsystem be very flexible (and expandable) in terms of how many terminals it could handle.

This would mean that such a special-purpose system, fitting between a general-purpose time-sharing system and a set of interactive display terminals, could be implemented economically to serve a trial set of terminals, and could then be expanded easily, as needed (with minimum cost penalty for not beginning with the final capacity).

3. Proposal I II: Film Output System

This is a proposal for setting up a flexible, high-quality facility allowing automatic output from our 940 files onto film, in such a way that microfilm, microfiche, or print copy can be produced with a full range of integrated text and graphic content.

We are not seeking publishing-house quality in font, resolution, or stability. Such quality would be desirable in principle, but its incremental value for our purposes does not seem to justify the associated incremental costs for acquisition and operation.

We do seek a quality suitable for technical reports and documentation, but the most important added feature we seek is the capability to produce (under program control) arbitrary characters and figures at arbitrary locations on a page and to produce all the pages of a document in automatic succession.

We want to be able to handle all of the content of the documents (except photographs) within the computerized system, including the processes of composition, study, modification, and output.

We also want to use this graphic freedom of page composition to explore new techniques of hard-copy presentation for the types of material we will handle in the NIC and in our other AHI work.

We feel that it will be very valuable to the experimental scope and progress of the network if critical documentation can be flexibly and rapidly updated and then rapidly and cheaply distributed. We intend to handle text, graphs, line figures, tables, and equations.

Without the proposed equipment, we would plan on using a lower quality system that is available on a closed-shop basis within SRI. We have already begun developing the programming and conventions for using this type of graphic output.

Appendix C
THE NLS VECTOR PACKAGE

The vector package allows the user to create simple line drawings, with labels, as a part of his file.

Drawings may be output via the printer or via film; on output to other devices they disappear.

The following is an excerpt from the NLS User's Guide; the syntax of the commands is given in a modified Backus-Naur form, but this need not concern the general reader. This material is given here simply to provide a rough idea of Vector Package operation.

1. Vector Package Commands

a. Insert Commands

(iv) I nsert Vector

S yntax: i v (CA / B / CD) CA

Semantics: The bug mark is used to determine the endpoints of lines.

Each CA after the first determines a line.

Thus four CA's produce three lines, with line 1 meeting line 2 at the position of the second bug mark, and line 2 meeting line 3 at the third bug mark.

To "lift your pencil" and break the continuity of the lines type a "B" or a CD.

(il) Insert Label

Sy ntax: i l SPACE LIT CA CA

Semantics: After typing the label, hit a CA to attach the label to the bug. The next CA fixes the label in its current position on the screen (rounded off to the nearest position that can be output on the printer).

b. Move Commands

(mv) Move Vector

S yntax: m v (bug selection of vector) $(left mouse button) (bug selection of point)

Semantics: When the vector is selected, its ends are marked O and X. The end marked X will move to the point selected

and the end marked O will remain fixed.

Hitting the left-hand button on the mouse wll cause the O and X to be interchanged.

The bug is then moved to the desired point and a CA hit to select the point. The "X" end of the vector will move to this point.

(ml) Move Label

Syntax: m l [1] CA CA

Semantics: When the first CA is hit, the label [1] is attached to the bug and moves with it. The next CA fixes the label in the new position.

(md) Move Drawing

Syntax: m d 2$2(bug selection of point)

Semantics: The two selected points define a translation vector, and each component of the drawing is moved by this amount.

c. Delete Commands

(dv) Delete Vector

Syntax: d v (bug selection of vector) CA

Semantics: Select the vector to be deleted and hit a CA.

(dl) Delete Label

Syntax: d l [1] CA

Semantics: The label [1] is deleted.

(dd) Delete Drawing

Syntax: d d CA

Semantics: All vectors and all labels in the drawing are deleted. The command is used for starting over from scratch.

d. Spacing Commands

(sf) Spacing Off

Syntax: s f CA

Semantics: This will set a flag that goes along with the picture telling the display creation routines not to space the statements to leave room for this picture.

(sn) Spacing On

Syntax: s n CA

Semantics: This is the complementary command to spacing off. Since the flag is set for spacing on as the default option, this command is necessary only to change the flag back.

e. Other Commands

(t) Translate Vector

Syntax: t (bug selection of vector) $(left mouse button) (bug selection of point)

Semantics: This command is identical to mv in terms of actions by the user to specify which vector and which end.

The end marked X moves to the specified new position and the end marked O moves in such a way as to preserve the length and direction of the vector.

(v) Vertical

Syntax: v (bug selection of vector) $(left mouse button) CA

Semantics: When the CA is hit, the end marked X is moved horizontally so that the vector is vertical.

(h) Horizontal

Syntax: h (bug selection of vector) $(left mouse button) CA

Semantics: When the CA is hit, the end marked X is moved vertically so that the vector is horizontal.

(g) Grid

Syntax: g CA

Semantics: The grid provides the user a means to draw "pretty pictures."

All positions are rounded off to the points on the grid.

The grid also places lines going through grid points such that they can be output on the printer and still look like straight lines.

The grid is either on or off; after typing "g" to get "grid" in the command feedback line, a CA causes the grid to change state.

(a) Abort

Syntax: a CA

Semantics: Everything that has been done in the current instance of the vector package is thrown away, the command "Vector Package" is aborted, and it is as if the command had not been given.

(f) Finished

Syntax: f CA

Semantics: This command returns control from the vector package to NLS proper.

UNCLASSIFIED
Security Classification

DOCUMENT CONTROL DATA - R & D

Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author) Stanford Research Institute Menlo Park, California 94025	2a. REPORT SECURITY CLASSIFICATION Unclassified 2b. GROUP n.a.

3. REPORT TITLE

Study For The Development of Computer Augmented Management Techniques

4. DESCRIPTIVE NOTES (Type of report and inclusive dates)

Interim Technical Report covering period 8 Feb. 68 thru 8 Feb. 69

5. AUTHOR(S) (First name, middle initial, last name)

Douglas C. Engelbart
William K. English
David A. Evans

6. REPORT DATE 8 March 1969	7a. TOTAL NO. OF PAGES 82	7b. NO. OF REFS 4
8a. CONTRACT OR GRANT NO. F30602-68-C-0286 b. PROJECT NO. 0967 c. d.	9a. ORIGINATOR'S REPORT NUMBER(S) Interim Technical Report SRI Project 7101 9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report) RADC-TR-69-98	

10. DISTRIBUTION STATEMENT

This document is subject to special export controls and each transmittal to foreign governments, foreign nationals or representatives thereto may be made only with prior approval of RADC (EMIIH), GAFB, N.Y. 13440

11. SUPPLEMENTARY NOTES Monitored By Rome Air Development Center (EMIIH) Griffiss Air Force Base, New York 13440	12. SPONSORING MILITARY ACTIVITY Advanced Research Projects Agency Washington, D. C. 20301

13. ABSTRACT

This report deals with certain aspects of a continuing research effort on direct augmentation of human intellectual effectiveness by means of full-time use of special on-line computer aids. Areas of special importance in this report are development and maintenance of a computer facility centered around a time-sharing SDS 940 and the development of on-line computer aids for management.

DD FORM 1 NOV 65 1473 (PAGE 1)
S/N 0101-807-6801

UNCLASSIFIED
Security Classification

UNCLASSIFIED
Security Classification

14 KEY WORDS	LINK A ROLE	LINK A WT	LINK B ROLE	LINK B WT	LINK C ROLE	LINK C WT
Data Processing						
Storage and Retrieval						
Graphics						
On-Line Processing						
Interactive Graphics						
Digital Computer						
Automated Management Information Systems						

DD FORM 1 NOV 65 1473 (BACK)
(PAGE 2)

UNCLASSIFIED
Security Classification