STANFORD RESEARCH INSTITUTE
MENLO PARK, CALIFORNIA

March 1967

Interim Progress Report

STUDY FOR THE DEVELOPMENT OF HUMAN INTELLECT AUGMENTATION TECHNIQUES

By: D. C. ENGELBART W. K. ENGLISH J. F. RULIFSON

Prepared for:

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
LANGLEY RESEARCH CENTER
MAIL STOP 126
LANGLEY STATION
LANGLEY, VIRGINIA 23365

CONTRACT NAS1-5904

SRI Project 5890

Approved: TORBEN MEISLING, MANAGER
SYSTEMS ENGINEERING LABORATORY

J. D. NOE, EXECUTIVE DIRECTOR
ENGINEERING SCIENCES AND INDUSTRIAL DEVELOPMENT

Copy No. 120

ABSTRACT

This report covers the first year of a continuing project, at the eighth year of a growing, multiproject program that is exploring the value of computer aids to augmenting human intellectual capability. Outlined briefly are the background and the "bootstrapping" nature of the program (e.g., the report was produced with the experimental computer aids) and its status at the start of this project. Advances during the year were in the programming system (CRT on-line debugging, etc.), in both off-line and on-line computer-aided text-manipulation systems implemented on a CDC 3100, and in specifications and designs for a time-shared system to be implemented in the next year on an SDS 940, which will be committed to serving the on-line CRT consoles for this program. Experiences with usage to date lead to an assortment of future research possibilities for both the computer aids and their utilization.

CONTENTS

ABSTRACT iii

LIST OF ILLUSTRATIONS vii

FOREWORD ix

I INTRODUCTION 1

1 The AHI Program 1
2 Specific Features of Our Experimental Environment 1
3 Value and Cultural Perspective 4
4 Status at the Beginning of the Project 7

II OPERATING-SYSTEM DEVELOPMENTS 11

1 Introduction 11
2 Hardware 11
3 The Off-Line Text System 13
4 COPE 16
5 SNOBOL3 17
6 The On-Line Text System 17

III NEW-FACILITY PLANS 23

1 The New, Time-Shared Computer 23
2 Other Facility Equipment 24
3 Software Studies 30

IV EXPERIENCE AND PLANS 37

1 Experience With Our User Systems 37
2 General User Practise and Needs 42
3 Control Procedures 46
4 Specific Near-Future NLTS Possibilities 51
5 Future Plans 56

V CONCLUSIONS 60

BIBLIOGRAPHY 62

APPENDIX A STRUCTURING CONVENTIONS A-1

APPENDIX B EXCERPTS FROM EARLIER REPORTS B-1

APPENDIX C NLTS Photos C-1

ILLUSTRATIONS

Frontispiece	The On-Line Work Station	
Fig. 1	The 940 Computer Facility Configuration	25
Fig. 2	Work-Station I/O System	29
Fig. C-1	The Underside of the Mouse	C-3
Fig. C-2	View of the On-Line Display	C-3, C-5, C-7

FOREWORD

This report summarizes the status of the current principal project within a multiproject program at Stanford Research Institute, aimed at increasing the intellectual effectiveness of problem-solving human beings. The Augmented Human Intellect (AHI) Program has been in existence since 1959, and has had a succession of sponsors and projects (see below).

This report differs markedly from other technical reports. A glance at its pages will reveal many stylistic differences; not so readily apparent are the reasons for the differences and the methods by which the report was prepared.

Viewed as a whole, the program is an experiment in the cooperation of man and machine. The comprehensible part of man's intellectual work involves manipulation of concepts, often in a disorderly cut-and-try manner, to arrive at solutions to problems. A common denominator among many of man's intellectual aids (e.g., notes, files, volumes of reference material, typewriters, adding machines, etc.) lies in externally representing his concepts by symbols, and finding aid for their external (to him) communication and manipulation. We are seeking to assist man in the manipulation of concepts--i.e., in his thinking--by providing computer aids for his minute-by-minute manipulation of these symbols.

We come, then, to the basic and visible difference between this report and other SRI reports: For one thing, the entire report has been produced with the current computer-aided techniques. Certain features of these techniques should be noted:

Statements--be they subheads, phrases, sentences, or paragraphs--are numbered and presented in hierarchical order. These statement numbers are one "handle" by which a statement may be grasped for any of the operations performed on or off line (see Appendix B). References, which appear in the Bibliography at the end of the report, are shown in the text by a mention of their statement names--e.g., (English1)--rather than by the more familiar superscript notation.

Once a given word in the report had initially been entered into the computer file, it was never again retyped by a human--through all of the successive drafts and including the final typing onto mats. All such modification and retyping was done either by our on-line text-manipulation system (at the direction of a person sitting at a CRT-display console), or by our compatible off-line text-manipulation system (at the direction of commands, carried by punched paper tape to the system, as produced by a person sitting "off-line" at a paper-tape-punching typewriter).

FOREWORD

Under the current Contract NAS1-5904 (jointly funded by NASA and ARPA), we have extended the on-line system and developed a new version of the off-line system for the CDC 3100; incorporated COPE, a new assembler and on-line debugger, as our on-line operating system; and made considerable headway in designing a multi-CRT, time-shared, on-line system for the AHI Program's forthcoming SDS 940. Other projects supporting the program were the following:

(1) A project for the Air Force Office of Scientific Research [Contract AF 19(638)-1021], under which the basic conceptual work was done, as well as the first off-line text-manipulation work

(2) An internally sponsored project at Stanford Research Institute, under which an intermediate-state off-line system was developed

(3) A project for the Advanced Research Projects Agency (Contract SD-269), under which work on information structuring, basic working methodology, and the higher-level manipulation processes in the on-line system were done

(4) A project for the Electronic Systems Divison of the Air Force [Contract AF 19(628)-1088] which studied structuring and manipulating techniques for managing information (specifically, system-program design documentation)

(5) A contract with NASA (Contract NAS 1-3988) in which we studied and developed the display-control techniques that represent the operational foundation of the on-line system.

The AHI Program is located organizationally within the Systems Engineering Laboratory (managed by Dr. Torben Meisling), which, in turn, lies within the Engineering Sciences and Industrial Development Area (directed by Dr. Jerre Noe). Dr. Douglas Engelbart is Head of the AHI Program, assisted by William English.

Current contributing staff: Donald Andrews, Roger Bates, Patricia Conley, Charles Dawson, Martin Hardy, Elton Hay, Jon Hopper, Stephen Levine, Hanne Olofsen, Bary Pollock, Johns Rulifson, James Stein, Margaret Watson.

Recent contributors: Thomas Humphrey, Charles Kirkley, L. Peter Deutsch, Butler Lampson, Sandra Palais.

THE ON-LINE WORK STATION

SECTION I: INTRODUCTION

1 The AHI Program

1a As a formally identified activity at SRI, the AHI Program is eight years old.

1a1 The only really comprehensive status report to date is (Engelbart1), published in 1962.

1a2 Since then, three reports on specific developments within the Program have been issued: (Engelbart2), (Engelbart3), and (English1).

1b This report is aimed at giving a reasonably complete report of program status, which is in line with the fact that this project was set up to support the program's basic development.

1b1 However, the program now has too many facets, developed to too much complex detail, to attempt making this "status" report as comprehensive as the earlier one.

1c Since the 1962 status report, our progress has been mainly in what might be called the "instrumentation" for the program therein laid out.

1c1 That is, little explicit development has been added to the conceptual framework which was the prime topic of the early work.

1c2 The program has managed to follow the "Recommended Research" outlined in 1962 very closely--albeit with more frustrations and mistakes than somehow seem appropriate.

2 Specific Features of Our Experimental Environment

2a Our initial focus has been on computer-aided text manipulation. (By "text" we mean generally information represented by strings of characters. This includes mathematical equations, programming statements, etc.) There are several reasons for this:

2a1 Text is representative of our speech and much of our conscious reasoning about nontextual records; it is the basic fabric in which most of the interpersonal collaboration in system development work such as ours takes place.

2a2 Text is applicable as a representation of our thoughts and actions at all levels of our working system (e.g., from coding for the computer up to long-range planning for the research program). This promises us a comprehensive integration of our aids into our basic approach to exploring computer augmentation.

2a3 A coordinated, working system for usefully manipulating text is relatively easy to implement. With equivalent resources, a wider collection of useful working aids may be implemented for text than for graphics, for instance.

2a4 An effective system for handling the text of working records (planning, design, reference, etc.) will provide a sound structure in which later to embed manipulation techniques for other symbols, e.g., graphics, mathematics, and chemical formulas. (Except in unusual cases of specialization, instances of a professional person's usage of these symbols are actually quite isolated in the context of his total working system when compared to the "text" manipulation he does.)

2b Text Structuring Conventions

2b1 With the general exception of COPE source code, all of our text material is formatted in what we call our "structured statement" (or sometimes, "linked statement") form, to facilitate computer-aided manipulation and study.

2b1a It was part of our initial program conception that special structuring of one's working information would be important, and this form represents our current developments in this direction.

2b1b For some material, the structured statement form may be undesirable (e.g., the Forward of this report). In these cases, there are means for suppressing the special formatting in the final printout of the structured-text files.

2b2 The structure has a hierarchical (or "outline") form, which is used to organize statements -- the basic working modules of our text.

2b3 The ability to name individual statements and to form arbitrary "cross-reference" links between any two statements, when added to the basic hierarchical form, yields a general structuring capability that is quite flexible.

2b4 These structuring conventions are assumed as temporary--it is expected that they will evolve relatively rapidly as our research progresses.

2b5 Appendix A describes these conventions in more detail.

2b6 We use these conventions for writing not only our memos, reports, User's Guides, etc., but also our detailed hardware-design specifications, formal syntax definitions, and formalized (human) procedure descriptions; and we are designing our new compiler (MOL) to accept source code that is so structured.

2c Means of Manipulating Working Text

2c1 Two coordinated systems of text manipulation are used within the AHI Program at SRI.

2c1a One is the Off-Line Text Manipulation System (FLTS), which provides a means for harnessing computer aid for a user sitting off-line at a paper-tape-punching typewriter.

2c1a1 The paper tape is later processed by a computer program (batch processing) that operates upon the new text input, as well as upon any specified "old" or previously processed text, according to user-specified directions (commands) embedded in the paper tape.

2c1a2 FLTS may be used for composing and modifying new files, for adding to or modifying old files, and for merging and reorganizing several new and/or old files.

2c1b The other is the On-Line Text Manipulation System (NLTS), with which a user sitting in front of a CRT display gets immediate response to his key-stroke and pointing actions in terms of modification to the displayed text, or to the place in text which he wishes to view, or to the form of a view which he wishes to see.

2c1c Output text from either NLTS or FLTS can be manipulated further with the other system.

2c2 All of our text-manipulation techniques will work to some extent with normal "free" text, but are much more effective and powerful when used on text structured according to our linked-statement conventions.

2c3 The purpose of these systems is to aid their users in composing, modifying, and studying their everyday working text--with special attention given to the ease, flexibility, and power that make working records "plastic" enough to be kept up to date with current thinking and developments.

2c4 Special printing-control features have been developed to facilitate output printing for either system. Special control directives are embedded in the text, to be carried and manipulated as part of a file's regular content. These are recognized and interpreted during output. By these means, margins may be set, page headings may be established, etc.

2c4a As an example, with the exception of those pages containing illustrations, every page in any of our current reports was printed on its mat directly from computer output.

2c4b The system, in response to directives, automatically left page space for the pages which contain photographs.

2c4c One directive controls whether or not the output processor will print any of the directives--to enable final output to be free of these non-contextual codes.

2d Note: We are currently in the process of redesigning our systems to be transferred to an SDS 940, which will be devoted solely to the support of the AHI Program. All of our current systems operate on the CDC 3100, whose use we share with several other projects.

3 Value and Cultural Perspective

3a There is little question that high-capacity, reliable computer service will soon be generally available, upon instant demand, at the fingertips of almost any professional who can derive enough value from this service.

3a1 There are many people working at bringing down the cost of such service, which is important to its practical utilization.

3a2 However, it is with the value, to the user, of plentiful and immediate computer service that this project is principally concerned.

3a2a We wish to work intensively at learning how to take maximimum advantage of such service, exploring basic rearrangements in modes of thinking, in conventions for symbolizing, and in the whole spectrum of working procedures.

3a2b We seek to maximize the value, in terms of increased intellectual effectiveness, that may be derived by skilled users of computer-aid systems.

3a2b1 This will entail harnessing every sort of computer aid that might reasonably become practical within the next few years and balancing the selection of possibilities to be thus explored, with due consideration to competition among the possibilities for the resources required to experiment with them (with some means to judge the potential value).

3a2b2 However, we should also invest in some possibilities just to learn more about them, so as to make a better judgement of their potential value.

3b From experience in discussing such matters, it is known that among different readers there will be a remarkable variance in the significance attributed to the above statements, depending upon the vision which each will have as to the nature, variety, and fundamental impact of "computer aids."

3b1 It is likely that the vision evoked in almost everyone's mind (today) will be quite pallid compared to the reality of a decade or so from now, with regard to the richness and intimacy of the interaction and to its impact upon our basic thinking and methodological characteristics and upon the concepts, notation, and visual portrayals of our "working languages."

3b1a Some, of course, will have a much more pallid vision than others.

3b2 The evolution of what can rightly be called a new cultural perspective is a slow and painful process, for an individual as well as for his society as a whole.

3b2a The conceptual framework or "system" culturally ingrained into us all does not make the consideration of this "system" much freer than was the case in Gutenberg's day.

3b2b True, we have more apparatus within our systems today; but the bushman has bright glass beads now, without having captured the cultural outlook that reveals an industrial economy as a coordinated and powerful thing, where many tools and many, many concepts, customs, procedures, etc. are interactively involved.

3b2c And so with us today: our "systems" find an increased availability of bright and shiny items, but no cultural outlook that lets us see an "intellect-augmentation" (I-A) system as a coordinated and powerful thing.

3b2d Today's culture does not yet see its current I-A system as haphazard or primitive, nor appreciate that directed endeavor could transform it into something radically different and more effective.

3c In the AHI Program, we are struggling to explore new tools, and the "many, many concepts, customs, procedures, etc. ... interactively involved", 3c1 We must report that along with this there is a parallel struggle, real and surprisingly difficult--to develop the appropriate new cultural outlook. We can certainly sense that the possibilities we explore and experience are the precursors of things radically different and more effective; but the integrated sense of it remains elusive--there is much talk and much struggling with details, while the seminal concepts slowly percolate to the forefront.

3c2 It is an enlightening experience, and produces a more real feeling of kinship with those ancestors whom archeologists say took 30,000 years to evolve from the first use of animal-skin garments to a concept of clothing and a process of curing skins.

3c3 We humans operate well enough in the patterns ingrained by our culture, but to press the frontier of established pattern reveals very quickly the innate weakness and clumsiness of our mental grasp.

3d General Features of Our Research Activity

3d1 It is primarily oriented toward learning how to use computer aids to increase appropriate human capabilities, and it gives attention to the problem of reducing the costs of these computer services only as necessary for its own effective experimentation.

3d2 It takes a "system engineering" approach to this matter of "increasing the value of plentiful and immediate computer service;" i.e., besides concerning itself with the development of the various computer-aid techniques and operations, it treats in a coordinated fashion other significant aspects of the "user system" that it finds to be amenable to study and improvement. For example:

3d2a The nature and structure of concepts with which the user approaches his problems.

3d2b The nature and structure of the symbolic representations of these working concepts--such as formatting, foot-noting, cross-referencing, special tagging, charting, etc.

3d2c The way in which the user "encodes" his requests for service from the computer system, considering both the mental and the physical actions that are required to control the computer "helper".

3d2d The procedures and methods with which he pursues his goals, at every level of activity at which the system designer can practically make changes.

3d3 This system-oriented approach is also an "experimental" one, in which all of the varied developments are put to work in a coordinated real-life working environment to be tested and evaluated.

3d4 The value ascribed to new developments, and the system of priorities used to select the next implementation from among many possibilities, are both based upon the associated improvements to the system's capability--i.e., to the working effectiveness of the people in this experimental environment.

3d5 The users of this experimental system, i.e., the "subjects" of experimentation, are the group at Stanford Research Institute which is concerned with the study, analysis, design, and implementation of human augmentation systems. (We call this our "bootstrapping" feature.)

3d5a Thus, the aids developed and experimented with are those which promise to the "Bootstrap Group" the best payoff either in direct improvement of their ability to "study, analyze, and implement," or in new understanding toward that end.

3d6 Implicit in the above, but deserving explicit comment, is the evolutionary nature of the system growth that results from this approach. Developments of various facets of this system, as well as our means to study, analyze, design, and implement them, must all evolve together in a coordinated fashion.

3d7 Note: Fairly extensive excerpts from the program's early report (Engelbart1) are included in Appendix B. These were selected to bring out in some detail the major conceptual principles that guide the program.

4 Status at the Beginning of the Project

4a At the time the project began, we had just replaced our earlier CDC 160A with the present CDC 3100. Our new facility comprised:

4a1 The CDC 3100 computer, in the following configuration:

4a1a Memory: 16,000 words, 24 bit, 1.75 microseconds

4a1b Three I/O channels, one of which is compatible with the interface previously used on some of our equipment for coupling to the CDC 160A

4a1c Paper-tape I/O

4a1d Three magnetic-tape transports

4a1e One IBM 1311 disk file (2,000,000 character capacity)

4a1f One 150-line/minute printer

4a1g Punched-card reader, 1200 cards per minute.

4a2 Work Station

4a2a Special (SRI) interface equipment couples the 3100 to a display generator, and to the various keyboards and selection devices of the work station.

4a2b Before the project began, SRI had installed a Straza character generator, with a repertoire of 63 characters and a generation rate of the order of 100 kc.

4a2c Early in the project, SRI added a vector generator to our display-driving equipment.

4a3 Software for the 3100

4a3a CDC provides a FORTRAN IV compiler, an assembler for their COMPASS symbolic machine language, and the SCOPE operating system, under which programs in either language may be run. (There are, of course, other CDC software systems, but these were the only ones of concern to us.)

4b NLTS had been programmed for the 3100, with essentially the same internal organization and external functional features as had been implemented on the 160A (see NASA1).

4b1 This implementation was programmed in COMPASS.

4b2 The 3100 had considerably more room in core for working data--so we now could contain the whole of a 30,000-character working file in core--whereas on the 160A only some 3000 characters out of a 17,000-character working file could be held in core.

4b3 The response to many commands, when working on a large file, was thus noticeably faster.

4b4 Quite a few improvements in detailed software design were made in this reprogramming, but it was not a "redesign," since our basic need was to get a system operating as quickly as possible on the 3100. The 160A was to be removed, and we did not want to be without a usable on-line text manipulation system.

4c The structure-manipulation part of a new FLTS had been specified for the 3100, and was being implemented in FORTRAN IV.

4c1 The manipulation functions specified were essentially the same as in the system previously implemented in ALGOL on our B5500--see ESD1 for description.

4d The basic assembly-debugging facilities within the COPE system had essentially been put into operation. This version of the system was still written in COMPASS.

4e A SNOBOL3 compiler had been designed for the 3100, and was essentially programmed (in COPE) but not checked out.

4f XDOC: External-Document System

4f1 Since 1959, under various successive sponsors, we have been accumulating a file of "external document" (XDOC) citations and reprints.

4f2 The citations had been punched on paper tape as they accumulated, and we had about 2000.

4f3 Of the actual referenced papers, at least 75 percent existed as reprints or copies, stored in our files under their XDOC accession numbers.

4f4 The only index we had to this was a card file arranged alphabetically by author.

SECTION II: OPERATING-SYSTEM DEVELOPMENTS

1 Introduction

1a Effort on operating-system developments has tapered off considerably since October 1966, in favor of increased attention to the design of our new system for the SDS 940.

1b Our NLTS (including the printout system), as implemented on the CDC 3100, is described in extensive detail (particularly from the user-feature viewpoint) in a forthcoming Interim Technical Report (Engelbart4).

1c Our FLTS, as implemented on the CDC 3100, will be described in similar detail in a later report.

1d The COPE assembler and on-line debugging system, as implemented on the CDC 3100, is described in detail in a forthcoming Interim Technical Report (Hopperl).

2 Hardware

2a Movie Camera: An H-16 Bolex Rex Camera (Purchased as SRI Capital Equipment)

2a1 We modified the camera slightly and added a photoelectric pickup to provide two output pulses per exposed-frame cycle--one at the start of the shutter-full-open period, and one at the start of the shutter-full-close period.

2a2 We built a remote switching control so that, from the display station, one can switch the computer interrupt input between the standard 60-cps source and the 32-cps camera source.

2a3 This allows us to switch easily between ordinary viewing of the display (e.g., getting material ready for filming) and the actual filming operation, in which we need to synchronize display frames with the movie-camera shutter.

2b We constructed a new and better-looking binary keyset.

2b1 Unfortunately, the "feel" is not as good as that of the old model, so we are still using the earliest "breadboard" version.

2b2 We are going to have to learn more about what makes a good keyset "feel." A travel and pressure difference that was hardly noticeable to non-users made a significant difference in speed and accuracy.

2c Screen-Selection Experiments

2c1 Knee Control

2c1a We wanted to give a better try to the knee-control idea that was tested briefly over a year ago. One problem was that the mechanism was hard to adjust for different users and for different working positions of a given user.

2c1b We found an easy way to implement a more flexible version.

2c1b1 The rho-theta (radial-rotary) Grafacon transducer (which we had previously tested for horizontal, table-top coupling), was mounted on its side under the table.

2c1b2 A wire hook that could clamp lightly over one's knee (from the side) was attached to its slider arm.

2c1b3 Vertical knee motion now couples to the rotary potentiometer, and sideways knee motion couples to the radial potentiometer.

2c1c The control technique works, but it feels awkward and needs more testing. It should be evaluated further along with the nose pointer, etc.

2c2 Nose Pointer

2c2a One of the possibilities for new screen-selection devices that was considered during the 1961-65 NASA project was one operated by nodding or rotating the head (moving the bug by "pointing the nose").

2c2b An easy way to make a trial implementation was conceived and carried out.

2c2b1 Our old joystick mechanism was mounted on top of a construction-worker hard hat, with the stick pointing forward so that it could be moved up and down and side to side.

2c2b2 A long, light, stiff metal tube was fixed over this handle, extending from the hat (when the user sits in front of the display) through a swivel-slider holder centered over the top of the display housing.

2c2b3 Turning or nodding the head causes the joystick mechanism to produce the two corresponding potentiometer voltages, which are used by the computer to move the bug on the screen just as for the mouse.

2c2c The nose pointer has been given only cursory evaluation trials, but it does work.

2c2c1 A user has his hands free for keyboard action while he moves the bug on the screen with nose-pointing motions.

2c2c2 After a few minutes one's head motion seems to get a little jerky (especially on small horizontal motions); it becomes a little hard to make accurate character selections. We attribute this to cramping due to lack of practise.

2c2c3 The hat should be lighter, and should not sit so high on the head.

2c3 Improved control coupling between transducers and the bug (for mouse, knee control, nose-pointer, etc.).

2c3a We put together some potentiometers and some adjustable, floating power supples, with which the user can independently adjust the centering and bug-displacement scaling for each axis.

2c3b We are interested in experimenting with a smooth, monotonic functional relationship between nose-pointing velocity and the bug velocity.

2c3b1 In fact, one might introduce a dynamic relationship, involving more than just the first derivative of transducer position, that could give a ballistic response which, when mastered by means of suitable practise, might produce improved speed and accuracy.

3 FLTS, the Off-Line Text-Manipulation System

3a A set of text-editing features were specified which represented an extension and improvement of those of the original "Z Code System" that were part of the earlier FLTS. (See OSR2 and ESD1.)

3a1 These were added to the structure-manipulating specifications already being implemented. Implementation for both was in FORTRAN.

3b The entire system went on the air last June, and became usefully operational by August. It has been continually expanded and improved, and by now is a very powerful tool.

3c Basic Features

3c1 A newly typed entry may address any prior text, either within itself or within a previously generated file.

3c2 New statements within these existing structures may be inserted simply by typing an appropriate (perhaps interpolative) location number.

3c3 A prior statement, together with its entire substructure, may be moved to any part of the addressed information structure, merely by designating the new location number (perhaps interpolative) for the referenced statement. Its substructure will retain the same relative structural position to the referenced statement in its new location.

3c4 A prior statement, together with its entire substructure, may be deleted simply by referring to its location number (as it appears in the printout, regardless of intervening (but as yet unexecuted) specifications for structural modifications which may eventually result in a new location number being assigned to the referenced statement).

3c5 The text content of any statement may be replaced by merely designating its location number, then typing in the replacement text.

3c6 The text content of any statement may be copied, either as a new statement at a newly specified location number, or as material to be appended to the end of an existing statement.

3c7 Newly typed text content may be appended to any designated prior statement. This new text may include any of the editing directives described below.

3c8 Within any statement, embedded directives (special sequences of text characters) may be included (or appended, see above) to specify additions or modifications to that statement.

3c8a Using any one of three different means, a point in the text lying between the statement start and the occurrence of a given directive, may be specified.

3c8b Then one can either specify a string of characters that are to be inserted at the point, or one can specify that all of the characters from that reference point forward through the directive are to be deleted.

3c8b1 It is permissible to include within the characters thus being inserted at the reference point a directive to cause material preceeding that point to be deleted--thus effecting a replacement operation.

3c8b2 One means of specifying the reference point is by quoting a string of characters and specifying whether it is just before or just after the first occurrence of this string within the statement that is to be a specified reference point.

3c8b3 Another reference-point designation means is to specify the number of lines, words, and characters back from the directive to the reference point.

3c8b4 A third referencing means specifies first a line in the statement--allowing either a backward line count from the directive's location or a forward line count from the beginning of the statement--and then a word within that line by either a forward word count from the first word in the line or a backward word count from the last word in the line--and finally specifying a given between-character reference point as either a character count forward from the very front of the word or a character count backwards from the very end of that word.

3d Special Features

3d1 When addressing several prior files, whose range of statement numbers may overlap and thus threaten ambiguity in their referencing, an FLTS user may specify that for his current purposes, each of these files is to have a (different) specified sequence of characters prefixed to every one of its location numbers.

3d1a This allows him to establish temporary, unique referencing to each statement of each file, by prefixing that statement's location number with its file's assigned prefix characters.

3d1b He may thus assemble any number of files into one large structure, within which he may move, copy, delete, and insert statements and substructure sections to effect an arbitrary delete-merge-rearrangement.

3d2 A user may identify different intervals of his new input as pertaining to independent FLTS jobs, merely by specifying a unique "sequence number" for each job.

3d2a To switch his attention (i.e., his associated input material) to a different job, he need only specify the associated new sequence number, and begin typing material for the second job.

3d2b When this new input is subsequently processed by FLTS, the computer will initially isolate the various segments of input occurring between successive sequence-specification points.

3d2c Then, to assemble the relevant new input for each job, it will collect those segments that begin with the sequence-specifying number that corresponds to that job (as communicated to FLTS by the computer operator).

3d2d This provision allows a person sitting at his paper tape punching typewriter to shift his attention back and forth between various tasks with easy flexibility.

3e The off-line system will allow direct interaction with the disc files, allowing an off-line user to modify any file, or merge and modify any sets of files, without the bother of so much intermediate paper tape.

4 COPE

4a COPE was finished early in the project year, and has become the basic operating system for NLTS. Its on-line (CRT) debugging facility makes it very valuable to us.

4b A FORTRAN program was written which translates COMPASS source code into COPE source code--except for the macros. With the help of this program, we mapped the implementation of the COPE system into the COPE language, so that its subsequent debugging and extensions have been done with the aid of its own on-line debugging features.

4c Later in the project, COPE was expanded so as to be able to load and link to programs written in 3100 FORTRAN IV or in COMPASS.

4c1 FORTRAN and COMPASS routines, after translation into relocatable machine code by standard CDC translators, can be loaded by the modified COPE monitor, at the same time that it is assembling regular COPE code, so that processes written in these other languages can be mixed with COPE processes.

4d Also, a flexible overlay (from disk) system was added.

4d1 This allows free growth of the operating systems, so that we can develop large repertoires of commands and/or more sophisticated operations.

4d2 A COPE program may be written with parts of it designated to be stored on the disk and "overlaid" onto specified sections of core when called by another part of the program.

5 SNOBOL3

5a SNOBOL3 was initially expected to be used for programming our first off-line system on the 3100, and even earlier it was considered potentially usable for parts of the on-line system.

5b Although it was nearly ready for operation, we abandoned further development early in the project year. NLTS and FLTS both had to get underway (in COPE and FORTRAN IV, respectively), and we did not see enough relative value in continuing.

5c Very little project effort was involved--most of the development had been done earlier by SRI.

6 NLTS, the On-Line Text Manipulation System

6a Early in the project, we added a disk-file system allowing quick loading or storing of core-held file modules by the on-line user.

6a1 Some two and one-half million characters of textual data are now available (in file modules up to 30,000 characters) to the on-line NLTS user.

6a2 Previously, one had to type in the name of the disk file onto which he wished the current contents of core to be written (usually "over-written").

6a3 The new output-to-file feature is coupled to the optional convention of having the name of the file appear as the name of the first statement in the file (as held in core).

6a3a When the user specifies "rewrite disk," the computer will display on the screen, as the name of the file onto which to write out this core-held data, the word (if any) it finds in parentheses just after the location number of the first statement in this data.

6a3b The user may either accept this file destination (by hitting CA), or may replace it with a LIT entry, which costs him no more effort than it previously did.

6a4 This makes much faster and safer the frequent rewriting of working files.

6b Improved Display Feedback

6b1 Echo Register

6b1a In the left region of the CFL (the Command Feedback Line) there now appears a four-character register that always displays the last four characters entered by the user--either from the keyboard or from the keyset. There are special overbar and underbar combinations to indicate such as Command-Accept and Command-Delete operations.

6b2 Tracking-Spot Character

6b2a The character used for a tracking spot is automatically changed at different stages in the specification and execution of a command.

6b2b An UP ARROW character indicates that a select action is valid, a PLUS character that a select action is invalid.

6c The following new text entities facilitate designation of structure-alteration commands. (See Appendix B.)

6c1 BRANCH: composed of one "top" statement, plus all of the substatements that lie under it. For example, Branch 2a might be composed of Statements 2a, 2a1, 2a1a, 2a1b, 2a2, and 2a3. To designate this branch as a command operand, Statement 2a would be selected.

6c2 GROUP: a Branch without its top statement. From the prior example, the Group would be composed of Branches 2a1, 2a2, and 2a3. To designate the group as a command operand, any one of Statements 2a1, 2a2, or 2a3 would be selected.

6d Automatic Renumbering

6d1 Upon any Delete, Insert, Move, or Copy operation on Statements, Branches or Lists, the statement structure will be immediately and automatically renumbered.

6d2 It was found that sometimes a user would prefer not to have the renumbering to be automatic under certain periods of his editing. We thus added a pair of control statements that would respectively turn this automatic feature on or off.

6d3 There is a feature included in the specification procedure for each of these commands to allow the user to indicate the relative level that any newly located structure elements are to be assigned.

6d4 The new numbers will be assigned on the basis of relative level and ordinal position within the string of statements.

6e "Break Statement" Command

6e1 This allows a user to break an existing statement into two statements, with the computer automatically providing the appropriate interstatement gap and automatically giving the newly produced statement an appropriate location number.

6e2 The user specifies (with the "u" or "d" characters) what relative level above or below the parent statement the new statement is to have.

6f "Structure jumping" commands enable a user, by selecting a given statement on the screen, to jump directly to any one of the following structurally related statements:

6f1 List Successor--for example, from 2b directly to 2c, skipping over any intermediate substatements of 2b.

6f2 List Predecessor--for example, jumping directly from 2b to 2a, passing over any intervening substructure of 2a.

6f3 List Tail--for example, jumping directly from 2b to 2h, assuming that 2h was the last statement of the list 2a,2b,....,2h (the sublist of Statement 2) of which Statement 2b is a member.

6f4 List Head--for example, jumping directly from 2d to 2a, which is the head statement of the list 2a,2b,....,2h, of which 2d is a member.

6f5 List Source--for example, jumping directly from 2d to 2, which is the source statement of the list 2a,2b,....,2h.

6g Indirect Referencing, or "Marking"

6g1 This allows a user to assign (or reassign) any one of a number of special reference codes (or markers) to any given character point in his text structure.

6g1a Thereafter (until that marker is reassigned to another statement) he may use this special (abbreviated) reference code to refer to that point for any command--especially such as Hop Place, Move Statement, etc.

6g1b For example, in a Forward Statement command designating the statement which he next wants to see at the top of the screen, he may substitute for the selection of a statement on the screen the designation of any of the marker positions. The computer will use this marker position for the command operand, and thus present a new display in which the statement at the top of the screen is the one in which the associated marker was positioned.

6g2 This is to provide for very quick hopping back and forth between these "marked" points--more or less paralleling the situation at one's desk when he places various pages about his work surface for ready visual reference.

6g3 It also provides help for the process of collecting items into one location, as one scans through the rest of his material, by allowing an abbreviated reference to that location in a Move or Copy command.

6h Modifying the displayed "view" via two view parameters.

6h1 Level-Clipping Parameters

6h1a This allows the user to specify a lower level to which the computer limits the statements displayed--which allows scanning over the text structure and seeing for instance only the first- and second-level statements (i.e. 1, 1a, 1b, 2, 2a, 2b, etc.).

6h2 Statement-Truncating Parameters

6h2a This allows the user to limit the display of each statement in a chain of statements to a specified number of lines. This permits quick, superficial scanning of many more statements per frame.

6h3 An initial straightforward means to specify level or truncation numbers was to use a separate, special command for each. For each command, a SPACE clears the old setting (and establishes the no-clipping and no-truncating status if followed by the CA), and up to four digits may be entered to set a new number.

6h4 A technique for quick level-truncation specification (LTSPEC) was added.

6h4a A special set of single-character codes is used just before the final CA (Command-Accept) action in any of the commands which can establish a new display position, to specify a change in either the level or the truncation parameter, or both.

6h4b The current state of each of these parameters is shown on the Command Feedback Line.

6h4c In moving to a new viewing place, several quick key strokes (often just one) added to the command specification serve to set up the new "viewing" parameters (i.e., level and truncation parameters).

6i Statement Freezing

6i1 This allows the user to designate one or more individual statements (as he scans along a given chain of statements) which until further specification are to be "frozen" on the display.

6i2 This means that they will be moved to the top of the display, and remain unchanged while the rest of the display space is used for normal scanning.

6j A "pattern-match" compiler-interpreter statement filter was also added.

6j1 This permits one to specify a content requirement which will be used by the computer to decide which statements are to be displayed.

6j2 The user specifies this requirement in the form of a formalized expression, in a special content-specification language, as part of a regular text statement in his file.

6j2a The language is quite flexible, and not only allows for combinatorial (AND, OR, NOT) requirements on the existence of specified character strings, but also can specify that one expression must occur within so many occurrences of a second--to allow, for instance, requiring "memory" to appear in the statement within four words following "computer."

6j3 The user then executes a "Compile Pattern" command, with this statement as a selected operand.

6j3a The compiler processes the expression, and produces a content-analysis process which can examine any statement and decide whether or not it meets the specified content requirement.

6j4 He can then execute a control command which uses this process to decide, from any subsequent viewing point, which of the statements following that point will be shown on the display.

6k We needed more control of the output formatting, for both NLTS and FLTS. Accordingly, we developed one generalized processor (written in FORTRAN IV) to serve both systems.

6k1 There are conventions for embedded directives that will cause changes in any of the many parameters--such as line length, line spacing, lines per page, etc.--as well as to designate such things as Start New Page Here, Establish Running Header (with content xxx), Start Numbering Pages (with first number to be xx), etc.

6k2 For check drafts, the user will usually want these directives to remain in the text on output, so that they can be proofed and modified along with the rest of the content. But it will be possibile to have a printout in which these directives are suppressed, as would be desired, for instance, in the final output of a report.

SECTION III: NEW-FACILITY PLANS

1 The new, time-shared computer

1a For the central computer, we have chosen the SDS 940 time-sharing system which we expect to be delivered about 4 June of this year.

1a1 In the process of selecting the 940, we evaluated other possible computer candidates, especially the CDC 3300 since it is completely program-compatible with the 3100 for which the present text-handling system is written.

1a2 It became apparent very soon that no suitable time-sharing software was available for any machine except the 940. We would have to develop whatever was needed for serving the multiple display consoles.

1a3 Acquiring the SDS 940, on the other hand, involves reprogramming our current user-system software, but as pointed out in Quarterly Progress Report 2, our user system design is due for an overhaul in any case. If we engage in any extensive reprogramming effort to get onto a time-sharing system, it is more in line with our research goals to spend the effort rebuilding our user system rather than learning how to write time-sharing software.

1b The 940 facility is shown in block-diagram in Figure 1. It has the following major components:

1b1 A 940 central processor with time-sharing hardware

1b2 Four 16K memory banks with a word length of 24 bits and cycle time of 1.8 microseconds.

1b2a Each of the memory banks contains a "MAM" providing a second port to memory for connection of I/O devices.

1b3 Teletype communications equipment for connecting to 16 direct-coupled teletypes.

1b3a Some of these direct-coupled lines will be modified here to accomodate dataphone terminals.

1b4 Three magnetic-tape units with 75 ips speed and up to 800 bpi density

1b5 A paper-tape station with 8-level reader and punch

1b6 Two "RAD" storage units.

1b6a These are fixed-head discs, each with a storage capacity of 500,000 words.

1b6b The RAD connects to the memory buss through an SDS I/O channel called the "DACC".

2 Other Facility Equipment

2a In addition to the SDS equipment on order with the 940, the facility will include peripheral equipment made by other manufacturers and some equipment designed and constructed at SRI.

2a1 It is recognized that problems of checkout and maintenance will arise in this kind of mix of equipment, but we have found that some of the SDS peripheral equipment offered is not at all suitable for our special needs. In the case of the displays and work-station I/O system, it was not practical to buy from a single manufacturer, even to our specifications.

2b The non-SDS equipment includes the following:

2b1 A Bryant Model 4061 Disk File

2b1a This disc was selected primarily for compatibility with the time-sharing system developed in Project GENIE at Berkeley, where extensive file-handling software is being developed.

2b1b Moreover, it is expandable in capacity (with the frame we are ordering) to approximately 64 million words, and it is significantly less expensive than the disc offered by SDS.

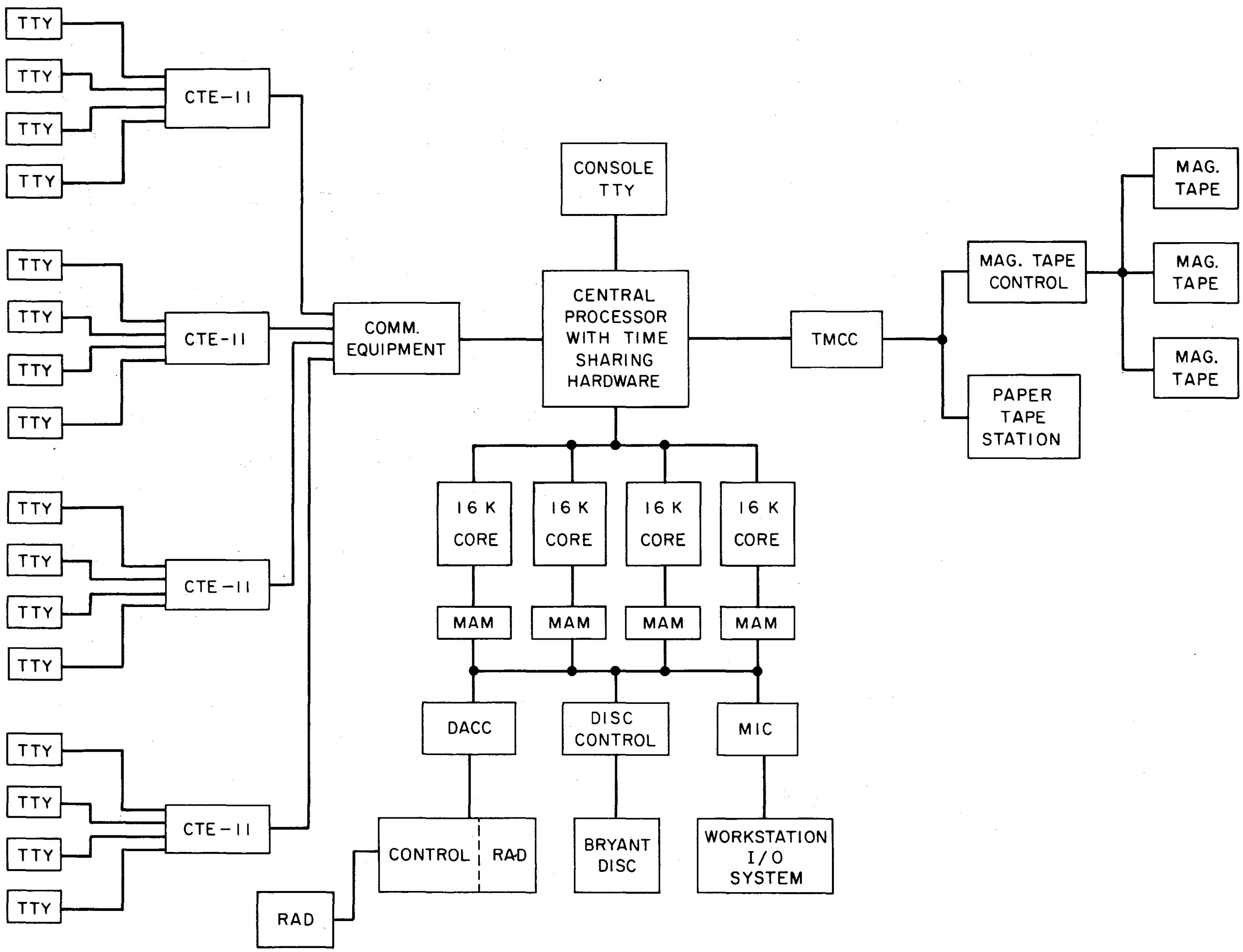
2b1c The disc will have a storage capacity, as formatted for our use, of approximately 32 million words, with a data-transfer rate of 40,000 words per second and average access time of 85 milliseconds.

2b1d It will connect directly to the 940 second port to memory through a controller-interface designed for this computer and supplied by Bryant.

2b2 A Model HSP-3502 Potter Line Printer

2b2a From the goals of our program, it is apparent that producing high-quality printed text in upper and lower case on a line printer would be extremely valuable.

2b2b We selected the Potter printer for the following reasons:



TB-5890-2

FIG. 1 940 COMPUTER FACILITY CONFIGURATION

2b2b1 It is a chain printer, which inherently produces better looking text than a drum printer.

2b2b2 It is significantly less expensive than other chain printers on the market, and from reports, is of high quality and reliability.

2b2b3 Project GENIE (at U.C. Berkeley) is also ordering a Potter printer, so that this choice, again, provides some compatibility.

2b2c The Potter printer will be bought with a full line buffer, and will be interfaced to the computer and controlled through the work-station I/O system, which is being designed and constructed here.

2b2c1 The line printer is not really part of the work-station I/O system, but much of the interface equipment that we are building for that system can be used for the printer as well.

2b3 The Work-Station I/O System

2b3a The work-station I/O system consists of the actual display work stations, and all of the hardware, back to the computer memory connection, necessary to serve them.

2b3a1 This system is being designed entirely at SRI. A major part of it will be constructed here, and other portions will be purchased to our specifications.

2b3a1a Again we realize the problems involved in undertaking a design of this magnitude, but we feel that the performance of this portion of the computer facility will have a significant effect on the kind of user service we can expect.

2b3a1b With the 940 time-sharing system, we suspect that without careful attention to the work-station I/O hardware the response would be limited by the CPU time available to the users for processing.

2b3a1c Therefore in the design of this sytem we have attempted to minimize the CPU time by making the system as automatic as possible in its access to memory, and by formatting the data in memory so as to minimize the executive time necessary to process it for the users.

2b3a2 The display system that we are now planning for the work station differs from that described in our proposal (Proposal for AHI Facility Expansion, SRI No. 67-10, 1 February 1967, prepared for Rome Air Development Center).

2b3a2a We originally proposed to implement a display system consisting of two 21-inch computer-refreshed (high-performance) displays and 10 storage-tube displays.

2b3a2b We now plan on a system of 12 computer-refreshed displays that will be viewed at the work stations by high-resolution (1029-line) closed-circuit television.

2b3a2c We have changed to the television system primarily because the Tektronix high-resolution storage tubes that we hoped to use are not available and may not be for another year.

2b3a2d We expect no significant cost difference, but there are other advantages and disadvantages to the television system.

2b3a2d1 On the positive side, we will be providing refreshed displays to users with the ability to change pictures much more rapidly than with storage tubes.

2b3a2d2 The most significant disadvantage is that more computer memory must be allocated to regenerating the displays. With our special interface design, we expect no significant loading on the CPU due to the additional refreshed displays.

2b3b The system is shown in block diagram form in Figure 2. It consists of the central and remote equipment to provide up to 12 active display stations located in offices throughout the laboratory area. It is divided into four functional groups:

2b3b1 Remote consoles

2b3b2 Closed-circuit television system

2b3b3 Display generators

2b3b4 Interface logic and controllers.

2b3c Each remote console will have a capability similar to that of the 3100 work station (see Frontispiece), including the following equipment:

2b3c1 A 17-inch television monitor

2b3c2 A 61-key typewriter-like keyboard

2b3c3 A binary keyset for general alphanumeric input

2b3c4 A mouse for controlling a cursor on the display

2b3c5 Provision for special input switches and audio-visual responses to the user (other than the display screen).

2b3d A high-resolution (1029-line) closed-circuit television system will transmit display pictures from the CRT monitors to television monitors at the work stations.

2b3d1 There will be 12 cameras in the system, each mounted in front of one of the 12 CRT monitors of the display generating systems.

2b3d2 A patch panel will be provided for connecting cameras to monitors with provision for feeding more than one monitor with the same camera.

2b3d2a When funds are available, we plan to replace this patch panel with a remotely controlled switching system so that camera switching can be done from the workstation.

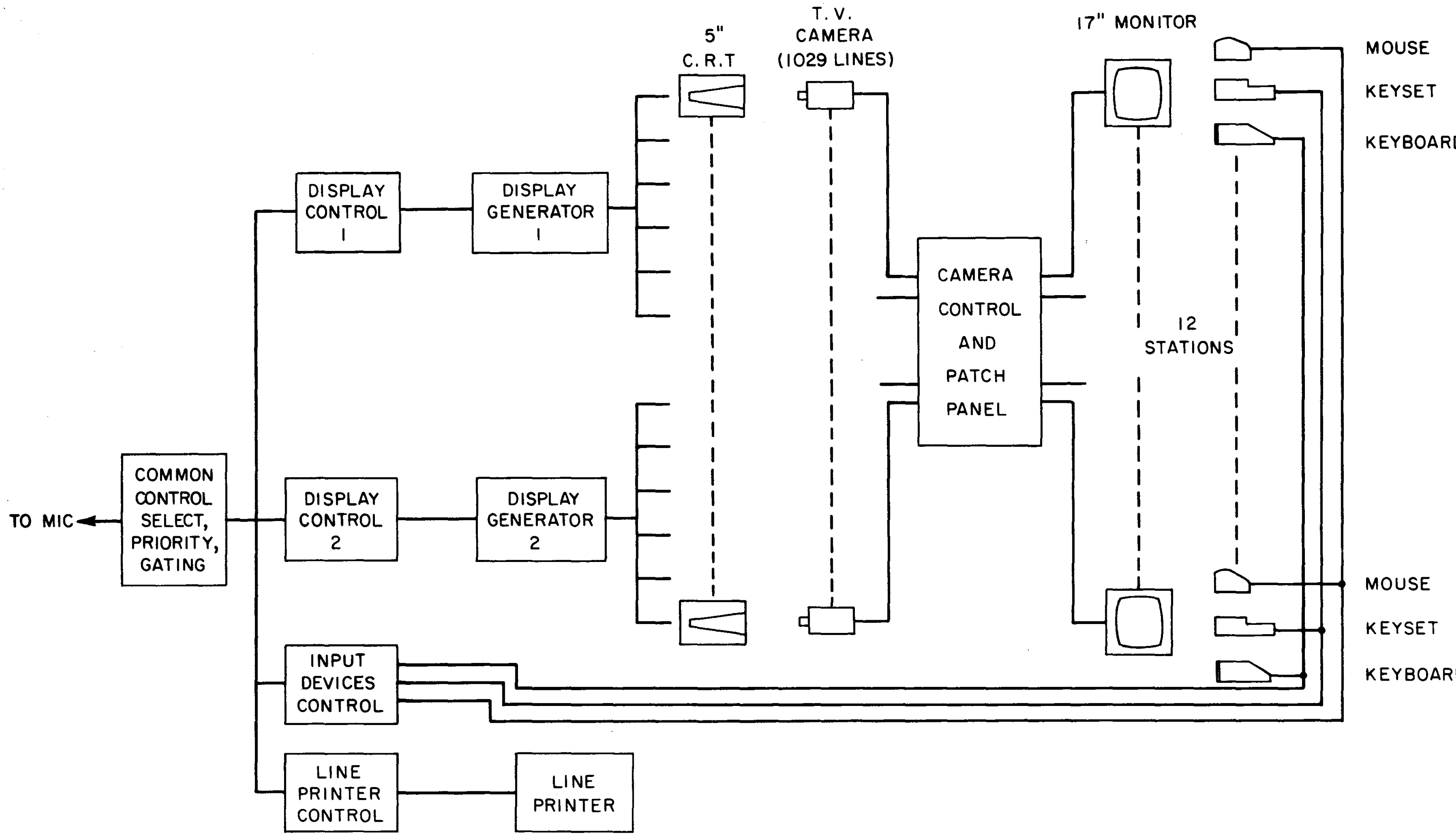
2b3e Each of the display generators will have general character-vector plotting capability. They will accept display instructions (beam motion, character writing, etc.) from the computer, and from them produce analog voltages to drive the CRTs.

2b3e1 Each will drive six 5-inch high-resolution CRT monitors on which the display pictures are produced.

2b3e1a The analog deflection signals from each display generator will drive all six of the CRT monitors in parallel, and selective unblank signals will determine which monitor is being written.

2b3e1b Character writing time will be approximately 5 microseconds, allowing an average of 1000 characters on each of the six monitors when regenerating at a 30 cycle rate.

2b3f The interface and controller group will accommodate two independent display systems, the console input devices (keyboard, mouse, etc., described above) and the line printer.



TB-5890-1

FIG. 2 WORK-STATION I/O SYSTEM

IT WILL HAVE THE FOLLOWING GENERAL FUNCTIONS:

2b3f1 Automatic direct access to the 940 memory through a Memory Interface Connection (MIC) connected to the second port to memory

2b3f2 Response to EOM instructions for program control of the work-station system

2b3f3 Response to program requests for status of the system (SKS instructions).

3 Software Studies

3a Most of our work to date has been somewhat informal, concentrating on exploratory effort rather than on production of code or formal algorithms. The major efforts have been in four areas:

3a1 Control metalanguage

3a2 Organizational and programming techniques

3a3 Development of MOL

3a4 Study of the 940 time-sharing system

3b The "Control Metalanguage"

3b1 In Section IV-3a, we discuss the "control language" with which the user and computer carry on a "control dialogue" -- where the human uses button or key actions and mouse movements to give control signals to the computer, and the latter uses various display signals as feedback.

3b1a The Control Metalanguage is designed for the sole purpose of describing the reactions by the computer that are specified for each possible action by the user.

3b1b Descriptions thus written are concise and unambiguous -- they are written in structured-statement form and are particularly suited for analyzing and modifying with NLTS.

3b2 This work, although it is still primitive in its development, has given us many insights into the organizational aspects of on-line communication. Only two of five major sections of the work are substainally complete.

3b3 The first of these--the state-machine deals with the control communication in a fashion which can be directly automated. The state-machine effort is substantially complete in its treatment of the devices it is set up to handle, except possibly for the display. The other devices (such as the keyboard, the mouse, and the pushbuttons) do not present any difficult problems.

3b4 The second section--the pattern-matcher section--is near completion. This part of the language deals with the selection of substrings of characters from larger strings. The language has the power to designate substring selection based on complicated contextual relations. In fact, all of the text-editing operations in the current NLTS represent a subset of this part of the control metalanguage.

3b5 We hope to develop the language to a point where the entire NLTS is a subset of our first SDS 940 service system. The language must be expanded to include new and different input and output media. It must be expanded to include much more in terms of display formatting (for example, graphics capability could be added).

3b6 We have formally described the syntax and semantics of our current Control Metalanguage, using in turn a description syntax suitable for driving an experimental syntax-driven compiler.

3b6a We are in the process of bootstrapping this compiler onto the 940 from an ALGOL version operative now on a Burroughs 5500.

3b7 This compiler will operate upon this formal description to produce a program (for itself) which in turn functions as a "control compiler" for the Control Metalanguage.

3b7a This control compiler is designed to produce a "state machine" which will be the work-station input-output control program.

3b7b A given version of this program will produce exactly those responses to each of the user's sequential control actions that were designated by the particular control-language description (written in Control Metalanguage) fed to the control compiler.

3b7c With this facility, we expect to be able very rapidly and easily to add to or modify the control language with which a user interacts with the computer.

3c Organizational and Programming Techniques

3c1 This portion of the program has gone down many blind alleys but the final results are maturing into very promising methods for the construction of a large service system.

3c2 It became apparent shortly after the work began that the new techniques of compiler writing make it possible that when any part of the system is described in a formal fashion, that description can be compiled into code of some type.

3c2a This means that the system can be divided into procedures and parameterized to a degree never before possible. This would imply an overall organization and flexibility that could be invaluable to the program as a research project.

3c2b To this end we have attempted to formalize the relationships and structures that occur in the type of program we will be writing for the SDS 940. The terminology and representation techniques that were developed are used as working tools in the rest of the project.

3c3 For storage and retrieval of data we have found some new and interesting algorithms as well as some different approaches to study, for the problem of establishing and retaining relationships within the data.

3d Development of MOL

3d1 The MOL is a "machine-oriented language" that will be used to write the rest of the system. "Machine-oriented" means that although the language looks like ALGOL, and embodies many of the syntactic features of ALGOL, it is heavily oriented toward the SDS 940 hardware. In fact, it may be used as a high-level assembly language.

3d2 When we were sure that the programming effort for the Bootstrap Program would proceed to the 940, we began a search for an appropriate coding language.

3d2a As we considered various languages we attempted to find one which would be internally compatible with the proposed service system, which would integrate into the structuring conventions of the proposed system, and which would lend itself to the debugging power available to on-line CRT terminals in a time-sharing system.

3d2b Few sections of code in the current CDC 3100 system (or any forseeable system) are critical to the performance of the system to the extent that the code could not be produced by a compiler. Having a large system written in a high-level language seems to be the only way to satisfy the above goals.

Our problem then, was to find a language that allowed one to write system programs, that produced reasonable code, and that could be supported and modified when errors were found.

3d3 The decision to develop the MOL took a long time. We did not want to isolate ourselves from the rest of the 940 programming community with a new language, but the MOL seemed to be our only solution in view of the drawbacks to the manufacturer-provided systems.

3d3a At one time it appeared that TYMESHARE would produce a system for compiling either ALGOL or FORTRAN. These were basically acceptable, but it then developed that all the code produced by this system would be interpretive (with unacceptable execution speed), and we therefore abandoned this possibility.

3d3b The next possibility was the standard SDS 930 FORTRAN II system. It was indicated that this system would be compatible with the time-sharing assembly-language debugging system. This would allow us to use FORTRAN II, supplementing certain sections with machine-language subroutines, and to debug with DDT, the time-sharing debugging system. However, we learned that the FORTRAN II system was being developed in such a way that only very primitive debugging facilities were to be made available.

3d4 We thus designed the MOL and have almost finished its implementation.

3d4a Under this system the code can be written in a very elegant high-level language. The MOL compiler translates this code to the standard time-sharing assembly code (ARPAS). This, in turn, is assembled and loaded under DDT, so that the standard, full-scale debugging tools are available.

3d4b The problem of designing a language for system programmers is an interesting one. For certain critical sections of system programs, the compiler must produce optimized code. This means that the programmer and the compiler must work together, for the compiler alone cannot do the job.

3d4b1 To view the problem in a proper perspective, consider a compiler for a full ALGOL language that makes no effort to optimize code.

3d4b1a When one looks for a way to modify the system towards producing more optimized code, the most rewarding place to make simplifications is in the syntax and power of the language. If dynamic declarations, recursive subroutines, and call-by-name procedure parameters are removed from the language, the general reductions in compiled code is very significant.

3d4b1b Beyond this, extensive work on the compiler (employing heuristics for optimization) or further restrictions to the language, will not produce proportionate gains.

3d4b2 Format and syntax for the MOL statements were chosen to be compatible with our text-structure conventions, with the feeling that the structure should help make clear the algorithmic structure within the program.

3d4b2a A result of these considerations is the CASE statement, which reduces the need for extraneous labels and complicated switches. There is also a simplified GOTO statement, for which the flow of control is clearer to a reader.

3d4b3 By including a small number of unusual new constructs, the MOL gains a great deal of power over machine-independent languanges. These new constructs give the programmer direct access to the registers of the machine and permit immediate and indirect addressing. This incorporates the programmer into the compiler process in such a way that, for the difficult sections of code, he can use his knowledge of the machine to write the exact code he wishes to have.

3d4b4 In order to choose syntax and conventions that will be more desirable in the long run, we have given preliminary copies of the MOL report to other 940 users. Not only has this produced good feedback concerning specifics within the language, but the concept of the MOL has been favorably received.

3d4b5 We expect eventually (although not in the immediate future) to extend the MOL so that the debugging of code may take place directly with the MOL code, and so that the separate assembly phase of the system may be eliminated.

3d5 In summary, the MOL has developed into a language which can be used on many different levels.

3d5a It can be used exclusively as a high-level language.

3d5a1 On this level the language has IF, CASE, WHILE, GOTO, CALL, and FOR statements.

3d5a2 It has arithmetic and logical expressions.

3d5a3 There may be subscripted variables.

3d5a4 There is block structure for grouping statements (but no dynamic declarations).

3d5a5 There may be procedures, and they may be compiled independently.

3d5a6 There is the facility for global (or NAMED COMMON) labels or identifiers.

3d5a7 There is the equivalent of the FORTRAN DATA statement or the ALGOL FILL statement.

3d5b The MOL can also be used as an intermediate language, using ALGOL like statements but manipulating bits within words or making direct references to the machine registers.

3d5b1 It is on this level that the infix operators for bit manipulation are used.

3d5b2 There are built-in functions for shifting which are oriented toward the 940 shift instructions.

3d5b3 There are also infix operators for indicating indirect addressing as well as immediate addressing.

3d5c Finally, the MOL can be used strictly as an assembly language, by writing assembly-language code within a pair of special braces (which direct the compiler to treat the string within the braces as code for the 940 assembler).

3d5c1 On this lowest level, the programmer may use the macro features of the ARPAS assembler.

3d5c2 In fact, he may do anything that one coding in the standard assembly language could do.

3e Study of the 940 Time-Sharing System

3e1 One person in the project has devoted almost all of his time, during the past few months, to the study of the time-sharing system so that we will have someone up to date in every aspect of its development.

3e2 So far this study has indicated that no major changes will have to be made to the time-sharing system.

3f The immediate goals of our 940 software effort are straightforward.

3f1 From this point on, we will code until we have a system up and running.

3f1a The first system is intended to be a preliminary one, lacking many of the desired features. It may not be as comprehensive as the current CDC 3100 NLTS.

3f1a1 Our plan is that it will be expandable and if we can establish a small system, and if the people on the project can begin to use that system immediately, we should be able to accelerate the succeeding evolution by means of "bootstrapping".

1 Experience with Our User Systems

1a General Experience

1a1 Beginning about October 1966, the number of our researchers actively using both text-manipulation systems has increased considerably.

1a1a This has been especially true since December 1966, when we became almost completely engrossed in the "paperwork" of writing specifications for our new system, and in documenting our old systems. Essentially everyone in the program has been thus involved, and has been doing all of this work by means of FLTS or NLTS.

1a2 Some Observations from this Experience

1a2a Appreciation of NLTS, and addiction to it, are generally very quick to appear.

1a2b People naturally prefer NLTS to FLTS, in that the operations are generally easier to specify--and more importantly, are executed immediately.

1a2c However, FLTS has proven very powerful and useful in its supplementary role--since NLTS availability has really been very limited (around 20 hours a week, distributed among 12 people).

1a2c1 We have come to realize that FLTS would be an important supplement for the new system, even though our availability of on-line CRT terminals will be much larger (an immediate jump in availability of on-line terminals by a factor of about 20).

1a2c2 FLTS will at least be very useful for enabling clerical staff to transcribe input material effectively; moreover during those inevitable periods when the system is down, users will still be able to be busy specifying additions or changes to their files.

1a2d The set of "tools" provided within NLTS, although quite rich and sophisticated by prior standards of computer-aided text manipulation, represents in our eyes only an initial step towards the type of system to which our bootstrapping activity should evolve within several years.

1a2d1 The evolution will be very much accelerated by enabling us to do essentially all of our daily work at a CRT console.

1a2d2 Something we have all realized in using the current NLTS is that even this "initial" set of tools is rich in possibilities for altering the various facets of our work, and that it will take literally hundreds of hours of serious utilization before we will have even calibrated the extent and quality of the changes in our ways of thinking and working which a thorough integration of these tools into our working methods would produce.

1a3 Our dedication to doing our work completely with our systems, while being very stimulating and exciting, has produced some very noticeable stresses.

1a3a Most of these seem to stem from the very limited availability of NLTS to each person in the program.

1a3b One too often gets trapped into having files stored on the disc which have been recently modified with NLTS, but for which one does not have a hard-copy printout by which he can specify further changes (it is extremely dangerous to use FLTS to specify changes in a file without knowing its current content).

1a3b1 As a result, we often have to produce hard copy after every NLTS session, since it is very hard to predict whether during the several days before one will get his second hour of the week, he will want to use FLTS to make modifications and need to know its current state.

1a3c Besides the burden of large quantities of printout, there is real difficulty in trying to coordinate one's work so as to make the best use of the rather stiffly scheduled times at which he may use NLTS.

1a3d All of these stresses should be reduced very significantly with the much increased on-line availability.

1a4 Bootstrapping

1a4a We find that the practice of bootstrapping as a research strategy produces a special focus on problems--tending to give us a feeling that while the problems we tackle may not be the best (relative to pursuing the goals of augmenting human intellect with the most payoff), they at least have a high significance.

place where he temporarily is storing such notes, to modify or add to any of them.

1b2b3 As any such miscellaneous thought develops, it is easy (and delightful) to reshape the structure and content of its discussion material.

1b2b4 It is also easy and delightful to see a number of initially disconnected notes mature to the point where they are ripe to be integrated under one "topical" heading.

1b3 This "being able to do something" about each little notion has profound effects:

1b3a Perhaps it is a personal problem, but in old-fashioned desk work, or conference work, there always seems to exist for me a large amount of "mind burdening" stuff that just can't "have something done about it at the time." E.g.: notions, thoughts, agreements with others, wondering if ..., a question for so and so when next we talk about ..., etc.

1b3b In me this builds up a tension that is disagreeable and fatiguing and that also tends to subtract considerably from any end-of-day feeling of accomplishment--there is so much hanging "in my head" left to be taken care of that it is difficult to relax, to think calmly and deeply, etc....

1b3c Interruptions of a given work task now needn't represent "holes in my day's productivity."

1b3c1 Since in even a short period I can add something to my working structure of information, a feeling that something was accomplished can be derived from really quite short interruptions about widely ranging matters.

1b3c2 And when I later shift my attention back to one of these matters, there is a true feeling that those prior short visits can be quite readily salvaged and integrated into the continually developing material.

1b4 I find that I can express myself better, if I can make all the little changes and experiments with wording and structure as they occur to me. [And he experiments a little with using structural decomposition of a complex sentence.]

1b4a I find that I write faster and more freely,

1b4a1 pouring thoughts and trial words onto the screen with much less inhibition, 1b An Individual Testimony

1b1 Note: Whereas "chopped-up" utilization of NLTS introduces a certain amount of stress to a user, the experience from the relatively few occasions of spending six or more hours in one working day using NLTS has always proved to be very rewarding and stimulating. Some feelings and thoughts in this regard, as written with NLTS by one user (Engelbart) at the end of an extensive working session (eight hours, on a Saturday) are quoted below.

1b2 One can almost always do something direct and satisfying about either the big or the little needs that arise:

1b2a His think-work at hand:

1b2a1 One's effort during any given small interval of time should contribute in some way toward the current working goal.

1b2a1a As part of our view of a whole day of professional work, we don't really seem to know how to think about the effect of such an interval--what to expect to have learned, to have decided, to have changed in our recorded working information, etc.

1b2a1b Yet, how to measure, to analyze, and to design (conventions, procedures, computer aids) for the activity of such parcels seems important toward improving the effect of the whole day.

1b2a2 Being able so easily and quickly to change phrasing and organization within a statement, or to change the structural relationship among statements, somehow gives the feeling of making steady and significant progress toward a desired goal, i.e. of being able to make each small interval count in an explicit way.

1b2b His miscellaneous notes and thoughts.

1b2b1 To accommodate and preserve a thought or piece of information that isn't related to the work of the moment, one can very quickly and easily insert a note within the structure of a file at such a place that it will neither get in the way nor get lost.

1b2b2 Later, working in another part of the file, he can almost instantly (e.g. within two seconds) return to the

1b4a2 finding it easy to repair mistakes or wrong choices

1b4a2a so while capturing a thought I don't have to inhibit the outpouring of thought and action to do it with particular correctness,

1b4a3 finding that several trials at the right wording can be done very quickly

1b4a3a so I can experiment, easily take a look and see how a new version strikes me--and often the first unworried attempt at a way to express something turns out to be satisfactory, or at least to require only minor touch up.

1b4a4 Finding that where I might otherwise hesitate in search of the right word, I now pour out a succession of potentially appropriate words, leaving them all there while the rest of the statement takes shape. Then I select from among them, or replace them all, or else merely change the list a bit and wait for a later movement of the spirit.

1b4b I find that,

1b4b1 being much more aware of

1b4b1a the relationships among the phrases of a sentence,

1b4b1b among the statements of a list,

1b4b1c and among the various levels and members of a branch,

1b4b2 being able

1b4b2a to view them in different ways,

1b4b2b to rearrange them easily,

1b4b2c to experiment with certain special portrayals

1b4b2c1 not available easily in unstructured data

1b4b2c2 or usable without the CRT display,

1b4b3 and being aware that

1b4b3a I can (and am seeking to) develop still further special conventions and computer aids

1b4b3b to make even more of this available and easy,

1b4b4 all tend to increase

1b4b4a my interest and experimentation

1b4b4b and my conviction that this is but a peek at what is to come soon.

1b4c And I find that I am more satisfied with the material that I develop.

1b5 Important to me is that

1b5a this increased "practice" with formulating thoughts and relationships,

1b5a1 different in nature from and greater in every-hour quantity than anything I have experienced before,

1b5b gives me a feeling that it really will increase the growth rate of my intellectual capability.

2 General User Practise and Needs

2a Working Files

2a1 There are perhaps four hundred working files stored on the several disc packs used among the group.

2a2 There has been no significant loss of information due to system failures.

2a2a "User errors" have cost us occasional significant losses. These tend to arise as a result of the high speed with which people can modify a file, set up new files, and delete whole files. Occasionally a person is moving "too fast" and before realizing the consequences executes some irrevocable "obliteration" operation.

2a2b With growing practice at this kind of "fast living," a user tends to use these very features of fast action to set up "backstopping" copies of recent versions of files upon which he is currently working.

2a2c This same practice is a standard protection means against system failures, and basically accounts for the relatively low losses suffered thereby.

2a2c1 A practiced user will still use the system to advantage even under conditions where (for instance) an intermittent bug may be causing crashes of the system as often as every five or ten minutes.

2a3 The practice of "backstopping" is greatly facilitated by having a really quick means for a user to send the current working version of a file out to replace the disc-held version (i.e., to become the new "backstopping" version). Total time for such an action is about three seconds for us.

2b The Practice of Documentation

2b1 There is a noticeable and generally acknowledged change among the researchers in the program in the manner in which documentation is accomplished.

2b2 During the last three months, almost every person has been doing his work essentially by the process of continually expanding and updating the associated documentation.

2b2a The improved facility for expanding and modifying working files has thus led to a willingness to do one's work this way.

2b2b We have all come to appreciate that "doing one's work this way" in turn is having a definite effect upon the way our projects develop, and the way in which we tend to interact among ourselves--both effects are considered very healthy.

2b3 There has also begun to evolve not only a number of special conventions accepted within the group for formatting or handling the files, but--more significantly--a recognition of the richness of future needs and possibilities in this respect.

2b4 This experience has produced a very definite orientation among us as to some of the sorts of changes (and value thereby derived) to expect in our working methods when we move to our new, much expanded system.

2c The Structuring of Records

2c1 The generalized hypothesis developed earlier (Engelbart1) (Engelbart2) about the value of introducing explicit structure to one's working records as a means of deriving more aid from a computer, seems pretty much to be affirmed by our experience.

2c1a Indeed, it is likely that there will be a steady increase in the degree and sophistication of explicit structuring within our working records.

2c2 Both FLTS and NLTS derive a considerable degree of effectiveness from the structuring--enough so that none of our practised users would be willing to forgo these conventions.

2c3 A general sort of need for additional structuring is often met where one would like to have more than one substructure for a given statement.

2c3a For example, we often find ourselves chopping up a complex statement into a sequence of subsubstatements, as a means of providing greater clarity in its expression.

2c3a1 This has a definite appeal as a means of providing "stronger" structural entities from which to form a structure of description or argument.

2c3b At the same time, however, there will often be a need for a sublist of "categories" which one would like to include in support of the given statement--but if a substructure was already being used to give syntactic clarity to the complex expression represented by the statement, we currently face an awkward problem for which we have no better solution than to join the statement back into one string.

2c3c A very similar situation exists for the "header" statement of a file (i.e., Statement 0).

2c3c1 We would find it quite useful to have Statement 0 be considered as the "source" statement of the List 1, 2, 3....

2c3c2 On the other hand, we also need an actual substructure beneath Statement 0 to carry descriptive information about the file.

2c3c3 Again, we have no ready solution available with the current structuring conventions.

2c3d In several of the existing formal systems for manipulating list structures, there has been recognized the general need for (at least) a double sublist:

2c3d1 One is for the normal list of structural entities, and one is a so called "description list" which serves to describe the parent or source entity "as an entity."

2c3e When initially establishing our structuring conventions, we abandoned the possibility of such double substructure in the interest of expediency.

2c3e1 One can always derive an equivalent effect for a given parent statement by giving it two substatements: the substructure of one will serve as a descriptive structure for the parent statement, and the substructure of the other will serve as the "substantive" substructure of the parent statement.

2c4 After our multiconsole system becomes operative, we would like to search for some possibilities for accommodating this need. (The problem is mainly how to display the relationships--and worse, how to print the material.)

2d The General Problem of Evolving Records

2d1 When working with NLTS, one finds that his "work" actually consists of continually adding to and modifying the structure of his working record.

2d2 For the straightforward processes of rearranging and rephrasing that are involved in the evolution of such a record, the direct application of the NLTS command repertoire proves quite satisfactory.

2d3 However, one finds that there are other kinds of "evolution" going on for which better tools and procedures will be of value.

2d3a A new feature, process, or concept, etc. struggles forth; it is, perhaps, first referenced by some compound-word term, and later may be given a special name.

2d3b Both the referencing terminology and the scope of definition for this newly emerging concept will undergo evolution.

2d3b1 As a change occurs in either of them, a corresponding modification in terminology, organization, and content is often required throughout the associated record.

2d3c Early in the emergence of such a concept, one often realizes (or may be conditioned to realize) that it is one that is likely to evolve in either terminology or definition.

2d3d It seems likely that special tagging given to the term, and associated special identification introduced in the various parts of the working record where one is depending upon the definition of the concept, could lead to some useful procedures and computer aids to facilitate subsequent evolution.

2e The Process of Categorization

2e1 This is very a basic operation when one is developing one's own thinking material, or studying (and possibly integrating into his own work) the work of another person.

2e1a For instance, the matter of developing a useful reference record for the design of a system generally entails developing categories relevant to the needs of the later users in their referencing to the record.

2e1b Also, when working on some complex task, if one has done a proper job of categorizing the facets of the problem, he can turn to the solution of the subproblems with assurance that this subsequent attention to detail will be an efficient investment of his energies towards the solution of the larger task.

2e2 We are discovering that the use of NLTS gives us an exciting new capability of interjecting into one's record many detailed notions and notes--even though they arise while working on an unrelated facet of the problem--which often leads to finding a large number of miscellaneous notions on one's hands.

2e3 It is easy for us to tuck these notions into specified structural locations, which would make it easy to categorize and group them if it were clear what categories were relevant.

2e4 As a representative medium-level task, the matter of establishing a relevant categorization structure, and of integrating miscellaneous types of entities into it, is important for us to study and improve.

2e4a Here will be found a task in which an effective repertoire of concepts, conventions, computer-aided "tools," and procedural skills will evolve much more rapidly and efficiently by the kind of studies, innovations, and real-life utilization which we hope to use.

3 Control Procedures

3a Control-Language Development

3a1 Very important to the speed, smoothness, and effectiveness of one's goal-pursuing actions is the mix of "tools" (computer aids) which he has within easy reach.

3a1a In a physical workshop, this would refer to their particular size, shape and function, and would involve considerations such as how well the function of each individual tool fits the natural task breakdown in one's work; and for each tool, how easy it is to reach, to grasp, to manipulate, to adjust, and to keep out of the way when not in use.

3a1b In the computer-CRT environment, this relates to a similar set of features and considerations [see(Engelbart1)].

3a2 What we have come to call the "control dialogue" for our on-line work is the set of interchange messages going back and forth between the user and the computer.

3a2a There is constant feedback activity on the display, often involving many discrete display-change actions during the formulation and execution of one "unit message" (i.e., one command).

3a2b In this sense, it is much more intimate a dialogue than our concepts of "language" are used to dealing with.

3a2c To use the concept of "language" with complete relevance in our environment, we must also consider that the "encoding" of the messages be analyzed not just in "bits" of bandwidth, channel noise, etc., but in the correlation between the physical actions of the user with the associated discrete key actions etc.

3a3 We have long felt the need for making it easier to implement new tools, modify old ones, and experiment with variations in the "syntax" of the language between user and computer.

3a4 Although the control language for the 3100 NLTS is fairly sophisticated, a person who becomes skilled with it and who works very long in an intensive manner, comes to want even further refinement and sophistication in the "vocabulary".

3a4a For instance, one would like to be able to have more text-entity types which he can specify as an operand, thus: from the selected character C1 to the end of its word; from C1 to the beginning of its word; a word string, i.e., from selected word W1 through selected word W2; from W1 to the end of its phrase (or sentence, or statement); or from W1 to the beginning of its phrase, etc.

3a4b One would also like to be able to mix his entities within a command, thus: move word W1 to be inserted as a character string immediately following character C2, etc.

3a4c One would like to move or copy a specified text entity to replace a second specified text entity.

3a5 Besides a more sophisticated repertoire of commands, one would like a more efficient means of designating the operations to be performed.

3a5a We should expect from our procedure-analysis work to learn enough about the frequency and sequencing characteristics of the various command usages to enable us to reduce rather noticeably the number of human actions required to elicit a given amount of computer aid.

3a6 Such trends will tend to increase the size and complexity of the control repertoire, as well as to make increasingly abbreviated the encoding of the "messages" to the computer. That this will mean an increase in the amount of training required to reach peak performance is not in itself alarming to us.

3a6a On the one hand, our vocabulary will always contain a subset which would be as simple as possible to learn, and which would avail a user of the basic means of studying and modifying his files.

3a6a1 Thus a user will find his vocabulary and skill growing smoothly with his practice and understanding.

3a6b On the other hand, we are bent on discovering just how much value a skilled user can derive from such systems, and we feel that the decision as to whether or not there is enough value to warrant a given amount of training should be made after we have a set of users who have reached an adequate skill level and thus learned how much value was derived.

3a6b1 From our experience to date, we are certain that for any user to whom a system of computer aids is-going to provide day-after-day working aids (of the minute-by-minute, all-day sort), the threshold of acceptable conceptual and psychomotor training that may be involved in his becoming really skilled will be a good deal higher than what is commonly expected.

3a6b2 Certainly it will be acceptable if it is as difficult as learning to operate an automobile and to understand and observe the conventions and practices of legal and rational driving rules--which represents a greater set of concepts and skills to be mastered than we generally appreciate.

3a6c We recognize that within our own community there has developed a need for formalization of our training procedures.

3a7 Our Control Metalanguage and "control compiler" approach, as described in Section IV-3 for the planned 940 system, should significantly enhance our control-language experimentation.

3b User-Procedure Monitoring and Analysis

3b1 We developed a modification to NLTS which will use a (spare) magnetic-tape transport to record the sequence of users' actions during an off-line working session. For each recorded action, it identifies the action and gives the time to the nearest hundredth of a second.

3b1a We then developed an elementary analysis routine which produced usage statistics for specified actions or sequences of actions.

3b1b We did this to answer a specific question relative to system timing, in evaluating the proposed overlay-swap system which was eventually implemented on the 3100.

3b2 We have recently begun to extend this into a fairly significant (to us) study--which we expect to grow and continue as a regular part of our system study.

3b3 As more people have come to be regular users of NLTS, we are observing the following:

3b3a Some users do not seem ever to use some of the features of the system (often, different unused features for different people).

3b3b Some people seem to move faster than others.

3b3c Some seem to have developed a better command of their "vocabulary," in that they seem more quickly to be able to formulate an effective sequence of on-line operations to effect a given task.

3b4 We expect the monitoring and analysis to help us assess these differences in an objective fashion, and to help isolate needs and possibilities for improving the system.

3b4a For instance, we expect to be able to measure the speeds and skills invloved in the very basic operations rather directly.

3b4b We hope to be able to isolate the sequences of actions which comprise basic tasks (e.g., assigning a name to a statement, following a reference link, and then finding one's way back to the starting point) so that we can study their frequency of occurrence and the particular means by which different users go about performing these tasks.

3b4c We expect to be able to develop performance characterizations and ratings for these different skills and task sequences, by which we can perhaps develop motivation and/or training.

3b4d In (Engelbart1) (see excerpts in Appendix B), where a good deal of conceptual work was presented regarding the nature of one's "system of intellectual capability," the concept of a "hierarchical capability repertoire" was quite prominent in the discussion of how to improve this system.

3b4d1 It is very evident from the experience we have had to date with NLTS that after continued usage of such a system to do one's everyday work, the overall capability improvement that will be realized will arise from many changes which will have crept into the methods and conventions associated with the capabilities throughout the hierarchy.

3b4d1a For example, the capability for very fast editing and for specifying the content of the statements one wishes to have displayed to him are both needed in order to make quick, flexible, and effective use of the search feature.

3b4d1b Being fast and flexible with this searching technique lets one embark more quickly upon searches, and one tends to find numerous places in reviewing and modifying his files where he begins to use this searching aid.

3b4e It would be extremely valuable to our work to learn to analyze extensive procedure records in such a way as to bring out the actual hierarchical capability relationships, and not only to witness their evolution, but to be able to design for and accelerate that evolution in a more effective manner.

3b4f It is obvious that such ingredients as the intuitive leaps and heuristic decisions within the behavior of an intellectual worker are of fundamental significance. We are not assuming that the next year of learning how to analyze procedural records will produce the means to isolate and assess this type of high-level ingredient, but we do expect that our work will be an important step toward this sort of achievement--an achievement which must ultimately be pursued in any serious drive toward augmenting human intellect.

3b4g Incidentally, the study and analysis of these procedure records will itself be an activity for which we shall develop on-line working aids--which represents an important aspect of our "bootstrapping" approach.

4 Specific Near-Future NLTS Possibilities

4a Binary Keyset

4a1 For several years, there has been one habitual user (Engelbart) of our one-handed coding keyset.

4a2 With the increase in usage bringing several people to a skill level where they could do serious on-line work, they became motivated to learn how to use the keyset.

4a2a The advantages become very obvious after a bit of skill develops--to be able to have both hands concurrently active in the interspersed actions of pointing with one hand and indicating other command specifications with the other produces a significant improvement in speed and smoothness (even when the character entry rate with the one-hand keyset is relatively slow compared to the user's speed with a typewriter keyboard).

4a3 We observe that, once mastered, the keyset usage becomes a regular mode of work. These users are unanimous in evaluation of this usage as "very much worth the trouble to learn."

4a4 It seems to take relatively few hours (about 3) of concentration upon this usage to enable one to usefully begin to incorporate keyset entry into his command-specification input--and from there his skill steadily grows.

4a5 The currently used keyset is actually the very first experimental model we made. Subsequent trials have always produced models whose "feel" was not as satisfactory.

4a6 We would like to design keys whose touch is at least as satisfactory as the first model, but which provide a smaller and lighter keyset, as well as enabling us to experiment with sets of more than five keys.

4a7 Most of us who use the keyset regularly are quite sure that we can handle more keys and the resulting larger code set to definite advantage.

4b The Mouse

4b1 The new model of the mouse seems to be better than the original in terms of the way it feels when grasped and moved, the smoothness with which it may be moved, and the precision with which one can make rapid selections.

4b2 We would like, however, to give the user more flexibility in altering the scale factor by which an incremental factor of mouse displacement is projected as an incremental displacement of the tracking spot (bug).

4b2a This will require dealing with the incremental displacements of the mouse and bug in contrast to the current treatment of their "absolute" positions.

4b2b Several schemes are under consideration, but for initial experimentation it would be quite simple to accomplish this via software--although the added overhead burden to the time-sharing executive for thus handling a number of users would probably be too expensive.

4b3 We would like to make a simpler, more compact version of the mouse, but the general payoff to us at this time is relatively small.

4c Combined Mouse-Keyset Device

4c1 We have hopes of eventually making a lightweight combination of binary keyset and mouse, so that the hand which operates this device may both "point" and "talk."

4c2 Several of the keyset users feel that they could learn to handle such a device to very good advantage.

4d On-Line Command Macros

4d1 The way in which the 3100 NLTS is organized affords an opportunity (previously recognized, but not pursued) of relatively easily having NLTS digest a text string and interpretively execute operations corresponding to its response to such a string as entered by an on-line user.

4d1a To be effective, this requires establishing a convention whereby a special symbol represents the Command Accept action, and another special symbol indicates that the system is to wait for a bug select action by the user at this point.

4d1b These features will allow a stored, prespecified command sequence to "program" a set of operations in which are to be interspersed specified types of human control actions.

4d2 Along with the above modification, we would provide special text-buffer arrangements in which a number of these "macro strings" could be stored, independent of the current working file.

4d3 An addition would also be made to the user-code structure so that the user may specify the system's picking up and executing a macro string located either in one of the special "semi-permanent" macro buffers, or in a regular text statement within the current working file.

4d4 The user must also be able to replace the contents in the macro buffers with arbitrary strings developed by him with the normal composition and editing means of the system.

4e Pattern-Matcher Marker Setting

4e1 An addition to the pattern-matching programs, and to the syntactic conventions for the pattern-specifying statements, will enable a user to specify that during the pattern-matcher's analysis of each statement, if certain of the content conditions are met, the pattern-matcher is to establish temporary markers at specified points.

4e2 After analyzing each statement in this fashion, the pattern matcher will call for the execution of one of two prespecified macro strings, depending upon whether or not the total content specification has been met by this statement.

4e3 A number of very powerful and useful processes will thus become available to us (and many more will undoubtedly be discovered). Two examples follow:

4e3a For each statement that meets a given content requirement (e.g., is of a given level, belongs to a given branch, or contains two occurrences of the word "display," etc.) replace any occurrence of the string "vendor" with the string "manufacturer."

4e3b If a statement is found containing a link of type "xyz," then add this statement to the freeze list and hop to the referenced statement to continue the scan.

4f Cross-Reference Linkage

4f1 We have heretofore been using cross-referencing conventions based upon giving a statement a name and producing cross-references by explicit use of that name, or referring to the location number of a given statement.

4f2 These two different ways of referring to a statement are respectively termed "reference by identity" and "reference by location."

4f2a It has been implicit that if a reference is cited with a location number, it indicates reference by location, whereas a reference by identity is generally cited by use of statement's name.

4f2b When working with NLTS, a user finds this convention reasonably satisfactory, since he may use either a location number or a name to hop to the referenced statement using either a Hop Name or a Hop Place command.

4f2c For someone reading printed material, it is relatively easy to turn to a given location number, but it is generally difficult to find a statement by name.

4f3 On the whole, these conventions have proven very useful, but we would like to modify them as outlined below:

4f3a Besides "pointing" to a statement as a whole, we would like, upon occasion, to be able to refer to entities within a statement--e.g., a character, word, etc.

4f3b As well as being able to refer to any material within the referencing statement, we would like to be able to refer to material in any other file in the store.

4f3c At a point to which a cross-reference is cited, we should like more freedom in what is used for the printed identification of the referenced item.

4f3d We would like, for instance, to be able to have the hard copy always cite a reference statement by location number, regardless of whether it is a reference by location or a reference by identity. The syntax of the referencing text should explicitly indicate the type of reference.

4f3d1 For an identity reference, using location number, there should be automatic updating of the location number whenever a change in the structure puts the referenced statement at a different location.

4f3e We would also like to make it easier to establish a link than to install names and type in the link words (as is now required).

4g Generalized Hop Commands

4g1 With NLTS the Hop Commands are used to follow cross-reference links. When supplied with an operand (a selected word from the screen, or a "literal" word typed in), these commands instantly move the user's view to the statement being referenced.

4g2 With slight additions to the programs that service the Hop Place, Hop Name, and Hop Link commands, we can derive a single Hop command which will better serve our needs.

4g2a To permit this, the composition of the word selected as the operand for each of these commands must differ in a unique way from those for the other two commands.

4g2b By adding some character-string analysis into the execution of the single Hop command, the computer can determine whether it is following a reference by identity or a reference by location.

4g2c The main effect will be to streamline the vocabulary and improve the user's control efficiency.

4g3 We plan to add a Hop-to-File capability.

4g3a In its execution, the computer will first clear the working space of the current file, and bring the designated file in from the disk.

4g3b With the addition of a straightforward syntactic convention to the way in which a reference term is specified, hopping to file could also be accomplished with the universal Hop command.

4g3b1 For instance, a link "see(delbox:)" would signify, by the presence of the colon, that it refers to a file named "delbox," rather than to a statement within this same file by that name.

4g3c We would also add the convention that one could refer to a given statement within another file by means of a compound reference--again with suitable syntactic conventions so that the universal Hop command will recognize the nature of the reference.

4g3cl Thus, "see(delbox:act)" would specify that this link refers to a statement named "act" within the file named "delbox."

5 Future Plans

5a General Research Plans

5a1 The general plan of our continuing research may be characterized as follows, assuming that we have a bootstrapping activity already underway whose people, tools, language and methods represent the system being discussed.

5a1a (E1) Establish, expand, and integrate a file of the needs and possibilities which come forth or are searched out during all phases of our work:

5a1a1 The needs are expressed as problems which impede some capability, and which seem likely to be solvable with reasonable effort.

5a1a2 The possibilities are feasible-sounding innovations which promise to increase capability, and which seem likely to be implementable at reasonable cost.

5a1b (E2) Establish, and continue to develop, a system model oriented to service our practical needs in developing the system in question.

5a1c (E3) Use the system in doing our everyday work, and use observation, measurements, and analysis of this performance in supporting Efforts E1 and E2 above.

5a1d (E4) Using the model of E2, evaluate and select from the file of E1 the next system modifications to implement.

5a2 In Effort E4 above, the general criterion we plan to use for selection is which possibility promises the most increase, within six months, to our capability for doing this kind of research.

5a3 During the next year we expect to progress, with numerous loops, through all four of the above stages.

5a3a Our initial 940 user system will be quite general and will be based principally upon getting our programming design records into linked-statement form and getting the service of our text-manipulation aids integrated into their design and modification processes.

5a3b We would expect to evolve a better-structured need-possibility file, and to establish basic system models. The concepts, needs, and possibilities accumulated to date will all be integrated, plus significant additions expected from our much-increased activity at the user-system level.

5a4 The scope of the file of Effort E1 and the model of Effort E2 will naturally grow to encompass more of our total system (that which augments us in our total project work) than the programming activity. This tendency will be fostered by a reasonable amount of specific attention to this end, and will naturally lead to the selection and development (in the activity of Effort E3) of other user-system features. (See the following section.)

5b User-System Development Possibilities

5b1 Programmer-Aid System (PAID)

5b1a Programming is a key capability in our experimentation. Thus, within our "bootstrapping" strategy, we initially concentrate upon improving it as a means of getting the most return in our increased capability from our early investment of research energy.

5b1b We are already in a very good position relative to assembly-language debugging--referring to the DDT system developed for the SDS 940 by Project GENIE.

5b1c DDT however, is oriented for Teletype users, and we will be using CRT displays. We will be developing an interface system to match the DDT services to our way of using displays.

5b1d The use of the MOL compiler, and MOL structured-syntax characteristics, will let us apply our structured-text representation and manipulation techniques to the task of writing, modifying, debugging, and documenting programs.

5b1e Beyond that, we hope (perhaps within the next year) to develop aids for higher-language debugging, by adding special features within the MOL compiler (probably still harnessing DDT to a considerable extent).

5b2 Research-Program Management Aids

5b2a A project supported by RADC will be concentrating upon the development and application of an exploratory user system designed expressly to aid us in the management of our own program activities.

5b2b The planning and coordination of our system implementations will be an important area for us to explore.

5b2b1 For instance, implementing the equivalent of PERT might be undertaken.

5b2c Developing a user system to support the analysis of the user-procedure records (see Section IV-3b) will probably fall into this project's activity.

5b2d We need to develop a model, probably based initially upon flow charts (or their equivalent) of the methods and procedures involved in the operation of the AHI research program.

5b2d1 If our bootstrapping development is to be significant, it must do something like this to get a grasp of the activities and capabilities which it is trying to augment.

5b2d2 This approach will start at the "high end" of the methodology spectrum, in comparison with the monitoring and analysis of on-line user's activities as discussed above--the two approaches can probably share techniques of procedure design and analysis, and should eventually merge.

5b2d3 We expect to establish special user techniques (probably with special computer aids) for developing and studying the models.

5b3 The AHI-Research "Intelligence" Problem

5b3a Note: This generally involves gathering and rendering useful to our pursuit the relevant information discovered or developed in the external world. For instance--

5b3b Working with external documents (as in our XDOC system):

5b3b1 Collecting and filing

5b3b2 Organizing and retrieving

5b3c Generating compendia, annotated bibliographies, etc. for our own use and for publication

5b3d Extracting, normalizing and integrating

5b3d1 Searching out, isolating, and extracting the facts, concepts, techniques, arguments, etc. that promise to contribute to one of the problem areas within the AHI research program

5b3d2 Normalizing--i.e. converting to common terminology, organization, and formatting--and entering into the computer files (in our structured-text form)

5b3d3 Integrating (along with our own contributions) into the evolving, unified structure of knowledge, principles, techniques, etc. representing the significant "state of the art" in research progress toward learning how best to augment the human intellect.

5b4 Training and Indoctrination

5b4a We need a better way to get an initiated layman to the point where he can make satisfactory use of our computer aids, doing types of work that represent meaningful contributions in his professional framework, without the operating actions getting in the way of his concentration on the problem.

5b4b This can be helped partly by maintaining reference documentation explicitly oriented toward developing within an uninitiated layman the necessary basic understanding and skill, and partly by developing better methods for instructing him and better procedures for keeping track of his state of understanding and skill.

5b4c We are hoping that both instruction and testing can be helped considerably with interactive computer aids, using the basic facilities of our on-line system.

5b4d The development and maintenance of the necessary "programmed" teaching material for computer-aided instruction and testing should be a rather natural task to which to apply our basic on-line user system.

SECTION V: CONCLUSIONS

1 The activity of the project involves a relatively large collection of innovations, implementations, experimentation, and analysis.

1a They are all highly interactive, and all in a state that can readily be described as "initial exploratory."

1b We cannot give conclusive assessments of any of these, but we can conclude our report by giving some relative evaluations on various aspects of our program.

2 "Bootstrapping," as a basic research strategy for us, is beginning to work.

2a During the past few months, a great deal of our daily work has actually involved the steady development and updating of working documents by means of NLTS and FLTS.

2b As to the value of the bootstrapping approach, it really seems to be paying off in terms of stimulation and orientation.

2c When a single person who meets a given frustration when using the system is also very aware of other needs and possibilities arising for users, analyzers, specifiers, and implementers of the system, there is a much increased integration flexibility, refinement, and speed of innovation.

3 The structuring conventions for our working text are of significant value. Added conventions of this sort will be encouraged.

4 The off-line text-manipulation system has proven very valuable in our past and current operational environment, where our on-line time is extremely limited.

4a We have reached no conclusion as to the value which FLTS will have for us when we have the much increased availability of our new 940 system.

5 Our on-line text-manipulation system is developed now to the point where it begins to offer something of significance to the experiment of evaluating what on-line work can really do to augment human intellect.

5a It will require utilizing NLTS for hundreds of hours per user before a true perspective of its value will develop.

5b Within that period, there is little doubt that the system will evolve considerably farther.

6 There is a very rich field of possibilities to be explored relative

to new concepts, conventions, techniques, skills, methods, etc. to be developed for users of NLTS (and its descendants).

7 The one-hand keyset, as used within our on-line working environment, is definitely worth the trouble of mastering for those who will devote any appreciable amount of their professional time to working on-line.

7a Extensions towards more keys and more sophisticated encoding are worth pursuing.

8 There will be a large (and complex) body of knowledge and skill to be mastered in order to capitalize effectively on the potential of real-time computer aids.

8a Our experience verifies that an ever more sophisticated interactive language will evolve within our program--and the reasons for this would seem directly applicable to any situation where it is expected that serious professionals will be doing significant portions of their daily work by these means.

8b We further realize, for our own community, that this complexity has grown beyond us to a greater degree than we have appreciated. We need to inaugurate a more formal approach to the indoctrination of new people to our conventions and techniques.

9 The studies and plans developed to date for our multi-CRT console system are that we should get acceptable service for six simultaneous users--and we hope for twelve.

BIBLIOGRAPHY

1 (Engelbart1) Engelbart, D. C., "Special Considerations of the Individual As A User, Generator, and Retriever of Information," Paper presented by Annual Meeting of American Documentation Institute, Berkeley, California, October 23-27, 1960

2 (Engelbart2) Engelbart, D. C., "Augmenting Human Intellect: A Conceptual Framework," Summary Report, Contract AF 49(638)-1024, SRI Project 3578, Stanford Research Institute, Menlo Park, California (October 1962)

3 (Engelbart3) Engelbart, D. C., "A Conceptual Framework for the Augmentation of Man's Intellect," in Vistas In Information Handling, Volume 1, D. W. Howerton and David C. Weeks, eds., Spartan Books, Washington, D.C. (1963)

4 (Engelbart4) Engelbart, D. C., "Augmenting Human Intellect: Experiments, Concepts, and Possibilities," Summary Report, Contract AF 49(638):1024, SRI Project 3578, StanfordResearch Institute, Menlo Park, California (March 1965)

5 (Engelbart5) Engelbart, D. C. and Bonnie Huddart, "Research on Computer-Augmented Information Management," Technical Report No. ESD-TDR-65-168, Contract AF 19(628)-4088, Stanford Research Institute, Menlo Park, California (March 1965)

6 (English1) English, W. K., D. C. Engelbart and Bonnie Huddart, "Computer-Aided Display Control," Final Report, Contract NAS 1-3988, SRI Project 5061, Stanford Research Institute, Menlo Park, California (July 1965)

7 (Engelbart6) Engelbart, D. C., "Study For The Development of Human Intellect Augmentation Techniques," Interim Progress Report, Contract NAS 1-5904, SRI Project 5890, Stanford Research Institute, Menlo Park, California (March 1967)

8 (Engelbart7) Engelbart, D. C. and T. L. Humphrey, "User's Guide: On-Line Text Manipulation System," Interim Technical Report, Contract NAS 1-5904, SRI Project 5890, Stanford Research Institute, Menlo Park, California (March 1967)

9 (Hay1) Hay, R. E. and J. F. Rulifson, "MOL940: Preliminary Specification For An ALGOL-like Machine-Oriented Language for the SDS 940," Interim Technical Report, Contract NAS 1-5904, SRI Project 5890, Stanford Research Institute, Menlo Park, California (March 1967)

10 (Hopper1) Hopper, J. D. and L. P. Duetsch, "COPE: An Assembler and On-Line-CRT Debugging System For the CDC 3100," Interim Technical

BIBLIOGRAPHY

Report, Contract NAS 1-5904, SRI Project 5890, Stanford Research Institute, Menlo Park, California (March 1967)

APPENDIX A: STRUCTURING CONVENTIONS

1 Statements

1a The beginning of a statement is marked by :(FILESTART / LINEGAP) (HGAP / NULL) DIGIT, where the LINEGAP would belong to a preceding statement.

1b The length of a statement is arbitrarily limited to 2000 characters.

1c Its composition, as a string of characters, is quite arbitrary except for the special requirements for "location numbers," "names," and "links," which are described below, and except that it is not possible to include the statement end-begin sequence.

1d It is terminated by the sequence LINEGAP :(FILEND / (HGAP / NULL) DIGIT, where the (HGAP / NULL) DIGIT would belong to the succeeding statement.

1e Location Numbers

1e1 The first word of a statement is its "location number"; the first character of this location number is a digit. The location number consists of a string of digits and alphabetic characters, with no spacing gaps included.

1e2 A "field" in a location number is a continuous string of alphabetic characters or a continuous string of numeric characters, broken possibly by periods to indicate interpolative breaks (only found on newly generated material for input to FLTS).

1e3 The number of fields in a location number represent the "level of structural depth" of the statement.

1e4 The characters in a given field indicate the ordering on a unique list in the structure of statements.

1e4a For example: 2az follows 2z; 2b.d lies between 2b and 2c; etc.

1e5 The location number represents the unique location of its statement within the larger structure of statements.

1e6 Syntax

1e6a (NUMFIELD) = 1$DIGIT $(PERIOD 1$DIGIT)

1e6b (LETTERFIELD) = 1$LETTER $(PERIOD 1$LETTER)

1e6c (LOCNUM) = :((FILESTART / LINEGAP) (HGAP / NULL)) NUMFIELD $:m(LETTERFIELD NUMFIELD) $1:N(LETTERFIELD) :GAP ;

1e6d (DEPTH) = 1 + 2m + n levels ;

1f Names

1f1 A name may be associated with any given statement. This name is enclosed in parentheses, and is the first printing string to appear after the location number.

1f2 The length of a name is limited to 32 characters.

1f3 The first character of a name must not be a digit, but otherwise the choice and sequence of printing characters composing a name is arbitrary, but no GAPs may be included between the parentheses.

1f4 (NAME) = :((FILESTART / LINEGAP) (HGAP / NUL) LOCNUM GAP LPAREN) not(DIGIT) $31(PCHAR) :(RPAREN GAP);

1g Links

1g1 Special words called "links" may be included within statements, to establish cross-references to other statements.

1g1a As many links as desired may be included in any statement.

1g1b They may be located anywhere after the location number and name.

1g2 Syntax

1g2a (LINK) = :GAP LINKTYPE RPAREN REFNAME LPAREN :(PUNCTUATION / GAP);

1g2b (LINKTYPE) = $PCHAR ;

1g2c (REFNAME) = not(DIGIT) $31(PCHAR) ;

1g3 REFNAME is the name of a statement to which this LINK refers, and

1g4 LINKTYPE is the (arbitrary, user's choice) identification of the type of reference thus being made.

2 Lists of Statements

2a Any statement may have a "list successor," which is another statement. The sequential string of statements formed by the successor of a statement, by its successor, etc., until finally a statement is reached that has no list successor, is called a "list of statements."

2b The first statement on such a sequential list of statements is called the "head statement" of the list; the last statement on such a list is called the "tail statement."

2c A list may contain an arbitrary number of statements, but must have at least one statement.

2d For each statement in a given list, the last field of the location number indicates the statement's location in that list. Interpolative breaks may appear in a field of the location number; in this case the numbers indicate only the relative location number. A list that is in "clear ordinal state" will have no interpolative breaks in its last field; the last field then indicates the true ordinal location on the list.

3 List Structures

3a Various structural relations are (implicitly) provided for by the conventions described above: the sequential association of statements within a list, and interstatement linkages between any two statements.

3b In addition, there is "hierarchical" structuring of lists.

3b1 Each list of statements may be a sublist of one (and only one) statement; that statement is known as the "source statement" of that list. The location number of each statement on such a list will differ from that of its source statement only by the addition of one more field.

3b2 Any statement in a sublist may be the source statement for another sublist of its own, etc., to arbitrary depth. The sublist of a statement, plus the sublists of the sublist statements, etc., form the "substructure" of the given statement.

3b3 Statement ST2 is said to be a "logical successor" of Statement ST3 if there could exist a hierarchical structure of statements such that, by their location numbers, ST2 could succeed ST3 in the text. For instance, following a statement "2b3" one could logically accept only "2b3a," "2b4," "2c," or "3." The presence of any other location number than these on the next statement establishes a "logical break" in the text.

3c A "branch" is defined as a statement plus all of its substructure, and is identified by the location number of that statement, e.g. Branch 2f, etc.

3d A "group" is defined as all of the substructure of some statement. As it is used in NLTS, a given group, identified by selecting a given statement, represents the substructure of that statement's source statement--i.e. the selected statement is one of the highest-level statements in the designated group.

APPENDIX B: EXCERPTS FROM EARLIER REPORTS

1 The following material is extracted from earlier reports [see(Engelbart1) and (Engelbart2)] to aid in presenting the conceptual background within which this research is carried on.

1a Introduction

1a1 "By 'augmenting human intellect' we mean increasing the capability of a man to approach a complex problem situation, to gain comprehension to suit his particular needs, and to derive solutions to problems.

1a1a "Increased capability in this respect is taken to mean a mixture of the following: more rapid comprehension, better comprehension, the possibility of gaining a useful degree of comprehension in a situation that previously was too complex, speedier solutions, better solutions, and the possibility of finding solutions to problems that before seemed insoluble.

1a1b "And by 'complex situations' we include the professional problems of diplomats, executives, social scientists, life scientists, physical scientists, attorneys, designers--whether the problem situation exists for twenty minutes or twenty years.

1a2 "We do not speak of isolated clever tricks that help in particular situations. We refer to a way of life in an integrated domain where hunches, cut-and-try, intangibles, and the human 'feel for a situation' usefully co-exist with powerful concepts, streamlined terminology and notation, sophisticated methods, and high-powered electronic aids.

1a3 "Man's population and gross product are increasing at a considerable rate, but the complexity of his problems grows still faster, and the urgency with which solutions must be found becomes steadily greater in response to the increased rate of activity and the increasingly global nature of that activity. Augmenting man's intellect, in the sense defined above, would warrant full pursuit by an enlightened society if there could be shown a reasonable approach and some plausible benefits."

1a4 This is ... "a program aimed at developing means to augment the human intellect.

1a4a "These 'means' can include many things--all of which appear to be but extensions of means developed and used in the past to help man apply his native sensory, mental, and motor capabilities--and we consider the whole system of a human and his augmentation means as a proper field of search for practical possibilities.

1a4b "It is a very important system to our society, and like most systems its performance can best be improved by considering the whole as a set of interacting components rather than by considering the components in isolation."

1b Conceptual Framework

1b1 "The conceptual framework we seek must orient us toward the real possibilities and problems associated with using modern technology to give direct aid to an individual in comprehending complex situations, isolating the significant factors, and solving problems. To gain this orientation, we examine how individuals achieve their present level of effectiveness, and expect that this examination will reveal possibilities for improvement.

1b2 "The entire effect of an individual on the world stems essentially from what he can transmit to the world through his limited motor channels.

1b2a "This in turn is based on information received from the outside world through limited sensory channels; on information, drives, and needs generated within him; and on his processing of that information.

1b2b "His processing is of two kinds: That which he is generally conscious of (recognizing patterns, remembering, visualizing, abstracting, deducing, inducing, etc.), and that involving the unconscious processing and mediating of received and self-generated information, and the unconscious mediating of conscious processing itself.

1b3 "The individual does not use this information and this processing to grapple directly with the sort of complex situation in which we seek to give him help.

1b3a "He uses his innate capabilities in a rather more indirect fashion, since the situation is generally too complex to yield directly to his motor actions, and always too complex to yield comprehensions and solutions from direct sensory inspection and use of basic cognitive capabilities.

1b3b "For instance, an aborigine who possesses all of our basic sensory-mental-motor capabilities, but does not possess our background of indirect knowledge and procedure, cannot organize the proper direct actions necessary to drive a car through traffic, request a book from the library, call a committee meeting to discuss a tentative plan, call someone on the telephone, or compose a letter on the typewriter.

1b4 "Our culture has evolved means for us to organize the little things we can do with our basic capabilities so that we can derive comprehension from truly complex situations, and accomplish the processes of deriving and implementing problem solutions. The ways in which human capabilities are thus extended are here called agumentation means, and we define four basic classes of them:

1b4a "Artifacts--physical objects designed to provide for human comfort, for the manipulation of things or materials, and for the manipulation of symbols.

1b4b "Language--the way in which the individual parcels out the picture of his world into the concepts that his mind uses to model that world, and the symbols that he attaches to those concepts and uses in consciously manipulating the concepts ('thinking').

1b4c "Methodology--the methods, procedures, strategies, etc., with which an individual organizes his goal-centered (problem-solving) activity.

1b4d "Training--the conditioning needed by the human being to bring his skills in using Means 1, 2, and 3 to the point where they are operationally effective.

1b5 "The system we want to improve can thus be visualized as a trained human being together with his artifacts, language, and methodology.

1b5a "The explicit new system we contemplate will involve as artifacts computers, and computer-controlled information-storage, information-handling, and information-display devices.

1b5b "The aspects of the conceptual framework that are discussed here are primarily those relating to the human being's ability to make significant use of such equiipment in an integrated system."

1c Hierarchy of Intellectual Capability

1c1 "One can integrate his new ideas more easily, and thus harness his creativity more continuously, if he can quickly and flexibly change his working record.

1c1a "If it is easier to update any part of his working record to accommodate new developments in thought or circumstance, he will find it easier to incorporate more complex procedures in his way of doing things.

1c1b "This will probably allow him to accommodate the extra burden associated with, for instance, keeping and using special files whose contents are both contributed to and utilized by any current work in a flexible manner--which in turn enables him to devise and use even-more-complex procedures to better harness his talents in his particular working situation.

1c2 "The important thing to appreciate here is that a direct new innovation in one particular capability can have far-reaching effects throughout the rest of his capability hierarchy."

1c2a Note: We consider a "capability hierarchy" to represent the large collection (repertoire) of intellectual capabilities which an effective person must possess, as grouped into levels such that higher-level capabilities are derived from collections of lower-level capabilities.

1c2b "A change can propagate up through the capability hierarchy; higher-order capabilities that can utilize the initially changed capability can now reorganize to take special advantage of this change and of the intermediate higher-capability changes.

1c2c "A change can propagate down through the hierarchy as a result of new capabilities at the high level and modification possibilities latent in lower levels. These latent capabilities may previously have been unusable in the hierarchy and become usable because of the new capability at the higher level.

1c3 "To our objective of deriving orientation about possibilities for actively pursuing an increase in human intellectual effectiveness,

1c3a it is important to realize that we must be prepared to pursue such new-possibility chains throughout the entire capability hierarchy (calling for a 'system' approach).

1c3b "It is also important to realize that we must be oriented to the synthesis of new capabilities from reorganization of other capabilities, both old and new, that exist throughout the hierarchy (calling for a 'system-engineering' approach)."

1d The Basic Perspective

1d1 "Individuals who operate effectively in our culture have already been considerably 'augmented.'

1d1a "Basic human capabilities for sensing stimuli, performing numerous mental operations, and for communicating with the outside world, are put to work in our society within a system--an H-LAM/T system--the individual augmented by the language, artifacts, and methodology in which he is trained.

1d1b "Furthermore, we suspect that improving the effectiveness of the individual as he operates in our society should be approached as a system-engineering problem--that is, the H-LAM/T system should be studied as an interacting whole from a synthesis-oriented approach.

1d2 "This view of the system as an interacting whole is strongly bolstered by considering the repertoire hierarchy of process capabilities that is structured from the basic ingredients within the H-LAM/T system.

1d2a "The realization that any potential change in language, artifact, or methodology has importance only relative to its use within a process, and that a new process capability appearing anywhere within that hierarchy can make practical a new consideration of latent change possibilities in many other parts of the hierarchy--possibilities in either language, artifacts, or methodology--brings out the strong interrelationship of these three augmentation means.

1d3 "Increasing the effectiveness of the individual's use of his basic capabilities is a problem in redesigning the changeable parts of a system.

1d3a "The system is actively engaged in the continuous processes (among others) of developing comprehension within the individual and of solving problems; both processes are subject to human motivation, purpose, and will.

1d3b "To redesign the system's capability for performing these processes means redesigning all or part of the repertoire hierarchy.

1d3c "To redesign a structure, we must learn as much as we can of what is known about the basic materails and components as they are utilized within the structure; beyond that, we must learn how to view, to measure, to analyze, and to evaluate in terms of the functional whole and its purpose.

1d3d "In this particular case, no existing analytic theory is by itself adequate for the purpose of analyzing and evaluating over-all system performance; pursuit of an improved system thus demands the use of experimental methods.

1d4 "It need not be just the very sophisticated or formal process capabilities that are added or modified in this redesign.

1d4a "Essentially any of the processes utilized by a representative human today--the processes that he thinks of when he looks ahead to his day's work--are composite processes of the sort that involve external composing and manipulatinng of symbols (text, sketches, diagrams, lists, etc.).

1d4b "Many of the external composing and manipulating (modifying, rearranging) processes serve such characteristically "human" activities as playing with forms and relationships to see what develops, cut-and-try multiple-pass development of an idea, or listing items to reflect on and then rearranging and extending them as thoughts develop.

1d5 "Existing, or near-future, technology could certainly provide our professional problem-solvers with the artifacts they need to have for duplicating and rearranging text before their eyes, quickly and with a minimum of human effort.

1d5a "Even so apparently minor an advance could yield total changes in an individual's repertoire hierarchy that would represent a great increase in over-all effectiveness.

1d5b "Normally the necessary equipment would enter the market slowly; changes from the expected would be small, people would change their ways of doing things a little at a time, and only gradually would their accumulated changes create markets for more radical versions of the equipment. Such an evolutionary process has been typical of the way our repertoire hierarchies have grown and formed.

1d6 "But an active research effort, aimed at exploring and evaluating possible integrated changes throughout the repertoire hierarchy, could greatly accelerate this evolutionary process.

1d6a "The research effort could guide the product development of new artifacts toward taking long-range meaningful steps; simultaneously, competitively minded individuals who would respond to demonstrated methods for achieving greater personal effectiveness would create a market for the more radical equipment innovations.

1d6b "The guided evolutionary process could be expected to be considerably more rapid than the traditional one.

1d7 "The category of 'more radical innovations' includes the digital computer as a tool for the personal use of an individual.

1d7a "Here there is not only promise of great flexibility in the composing and rearranging of text and diagrams before the individual's eyes, but also promise of many other process capabilities that can be integrated into the H-LAM/T system's repertoire hierarchy."

1e Summary

1e1 "An hypothesis has been stated that the intellectual effectiveness of a human can be significantly improved by an engineering-like approach toward redesigning changeable components of a system.

1e2 "A conceptual framework has been constructed that helps provide a way of looking at the implications and possibilities surrounding and stemming from this hypothesis. Briefly, this framework provides the realization that our intellects are already augmented by means which appear to have the following characteristics:

1e2a "The principal elements are the language, artifacts, and methodology that a human has learned to use.

1e2b "The elements are dynamically interdependent within an operating system.

1e2c "The structure of the system seems to be hierarchical, and to be best considered as a hierarchy of process capabilities whose primitive components are the basic human capabilities and the functional capabilities of the artifacts--which are organized successively into every-more-sophisticated capabilities.

1e2d "The capabilities of prime interest are those associated with manipulating symbols and concepts in support of organizing and executing processes from which are ultimately derived human comprehension and problem solutions.

1e2e "The automation of the symbol manipulation associated with the minute-by-minute mental processes seems to offer a logical next step in the evolution of our intellectual capability.

1e3 "An approach has been outlined for testing the hypothesis of

Item 6a and for pursuing the 'rich and significant gains' which we feel are promised.

1e3a "This approach is designed to treat the redesign of a capability hierarchy by reworking from the bottom up, and yet to make the research on augmentation means progress as fast as possible by deriving practically usable augmentation systems for real-world problem solvers at a maximum rate.

1e3b "This goal is fostered by the recommendation of incorporating positive feedback into the research development--i.e., concentrating a good share of the basic-research attention upon augmenting those capabilities in a human that are needed in the augmentation-research workers.

1e3b1 "The real-world applications would be pursued by designing a succession of systems for specialists, whose progression corresponds to the increasing generality of the capabilities for which coordinated augmentation means have been evolved.

1e3b2 "Consideration is given in this rather global approach to providing potential users in different domains of intellectual activity with a basic general-purpose augmentation system from which they themselves can construct the special features of a system to match their jobs, and their ways of working--or it could be used on the other hand by researchers who wanted to pursue the development of special augmentation systems for special fields."

APPENDIX C: NLTS PHOTOS

1 The Frontispiece shows the NLTS work station.

1a A typical screenful of text is shown on the display.

1b Directly in front of the display is the typewriter keyboard normally used for extensive literal input, and also usable for designating commands.

1c To the right of the keyboard is a new model of our mouse, which a user can move over the surface of the work table to cause the tracking spot (bug) on the display to move so as to point to the displayed entities upon which he wishes to operate.

1c1 The rightmost of the three buttons on the mouse serves the double purpose of "select" and "command accept" (i.e., "execute").

1c2 The center button is used to tell the computer that instead of the above kind of operand selection, he wishes to designate an operand by means of an indirect reference to a "marker."

1c3 The lefthand button does not have a function--it is a spare.

1d To the left of the keyboard is the currently used keyset, with which a user may enter (with one hand, while the other is operating the mouse) command specifications or literal input--i.e., it may be used to enter anything that can be entered with the keyboard.

2 Figure C-1 is a closeup view of the bottom of the mouse.

2a It shows the two orthogonal wheels whose rotations respectively track the forward-backward and lateral displacement of the mouse. The third point of contact with the tabletop is the ball-bearing rider shown in the foreground.

2b Note: This is a recent redesign of our original mouse. The feel to the hand, the smoothness of operation, and the ease of making fine selections are all quite improved over the old model.

2b1 We are trying to make arrangements for this design to be produced for commercial sale by an appropriate manufacturer.

3 Figure C-2(a) shows a closeup of the display screen as it appears just after the file named "NLTSCOM1" was entered into the "work space" from the disc.

3a All of the text across the top line is "computer feedback."

3a1 The two, small indistinct blurs at the left show the current level (on the left) and truncation (on the right) parameters.

3a1a They both show "all," meaning that statements of all levels, and all lines of each statement, are being shown.

3a1b They will expand to normal size letters at a time, during command specification, at which the special LTSPEC characters may be used to modify the parameters.

3a2 Just under these two parameters is displayed the four-character "echo register." Currently showing the three letters "oml" and the underlined UPARROW symbol (which represents "command accept" action). These were the last four user actions in calling for this file to be brought in from the disc.

3a3 The "enter disc" words tell the user what command is still operative for the system, and the "NLTSCOM1" shows the literal operand that was just used to execute the command.

3b The rest of the screen displays the first part of the file.

3c Our convention is to use the name of the file as the name of Statement 0 for the file.

3c1 The use of our statement-naming conventions is seen in Statements 0 and 1b1, where enclosing in parenthesis the first word after the location number designates that word to represent the name of that statement.

3c2 An overbar over a character is our way of designating an upper-case character.

4 Figure C-2(b) shows the view generated from the start of the same file, but with the viewing parameters set to one level and one line.

4a A command to put Statement 1 at the top of the screen, and increase the level parameters to 2, has just been specified, and only needs a "command accept" action to complete its execution.

4a1 The last two characters in the echo register show respectively the select action (which is the same button depression as a "command accept" action) that was used to select Statement 1, and the subsequent "B" character which the user entered as the abbreviated LTSPEC action to specify that the level parameter was to be increased by one.

4a2 The view parameters as displayed represent those which will be in effect when the next view is generated; i.e., they provide

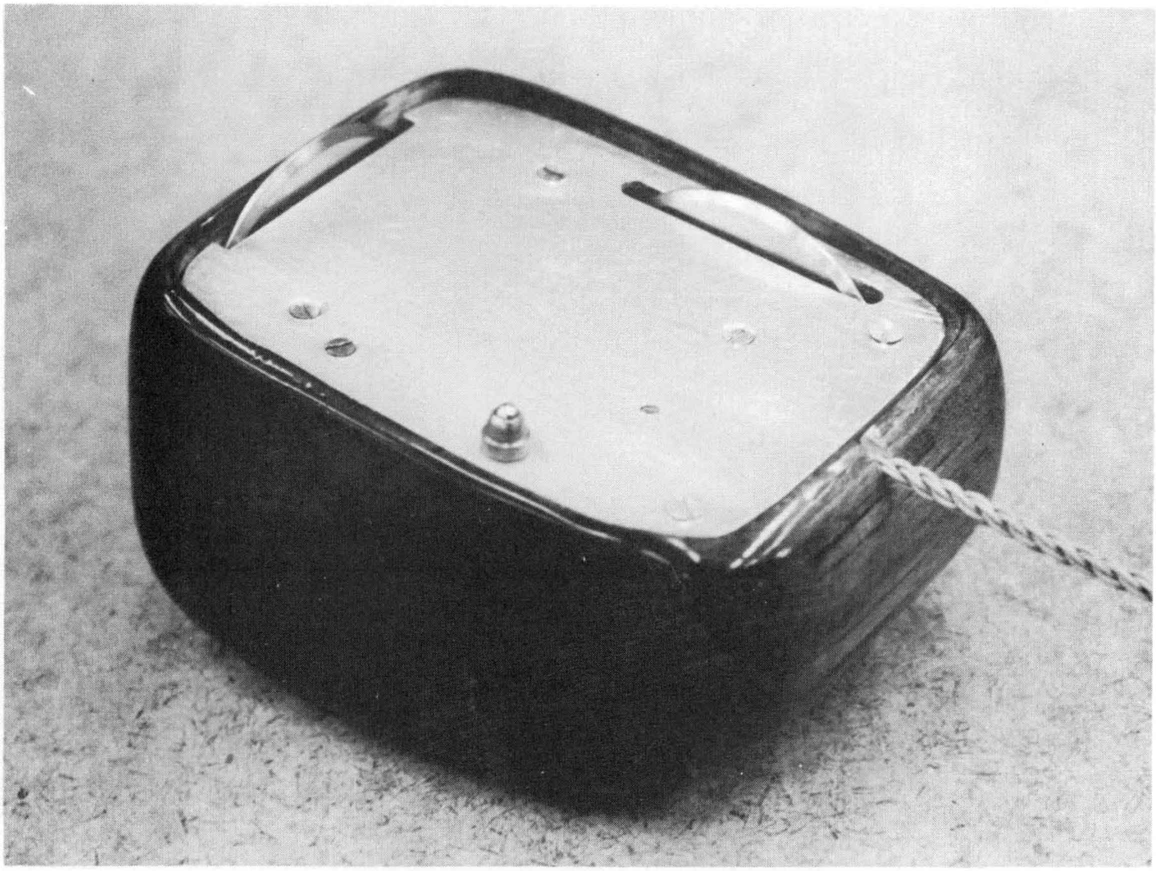
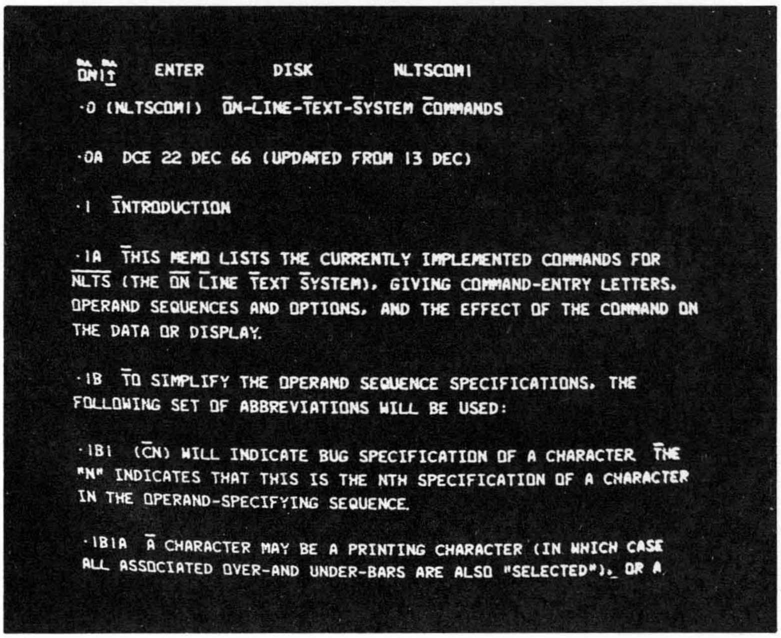


FIG. C-1 UNDERSIDE OF THE "MOUSE"



(a)

FIG. C-2 REPRESENTATIVE VIEW OF THE ON-LINE DISPLAY dynamic feedback to the user as he is in the process of specifying the new parameters.

4a3 The underline under the "I" in Statement 1 represents feedback that shows the user that this was the character which he selected--and since it is a "statement-entity" command, the computer deduces that it is Statement 1 which the user is designating as the one he wants placed at the top in the process of scanning forward within the file.

4a4 The bright spot above the underline, covering the bottom part of the "I," is the tracking spot--displayed as a PLUS character at the moment, since it is not at this time acceptable for the user to try to make another selection.

4b Note that this "view" provides a natural "table of contents" for the file--derivable with only a few quick actions by the user.

5 IN Figure C-2(c) a more detailed "table of contents" view of the file is what the user saw about 0.6 seconds after he subsequently pushed the "command accept" button.

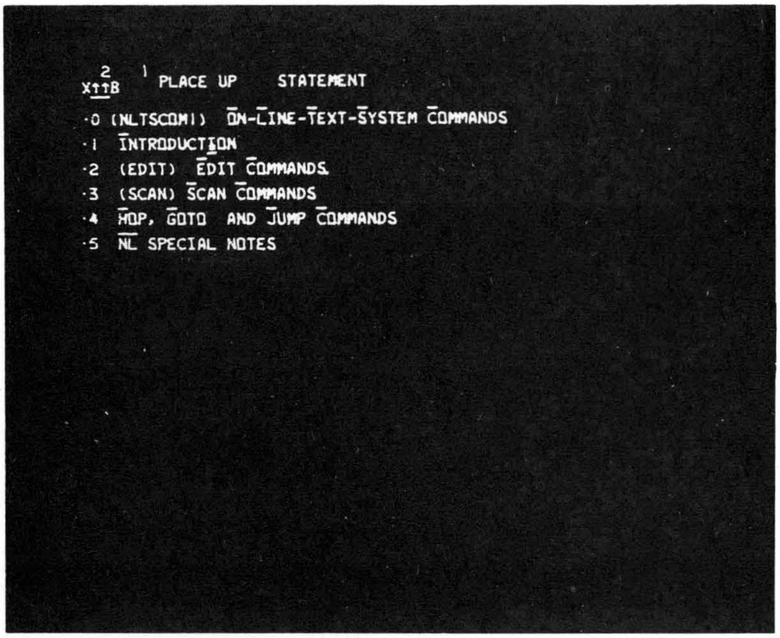
5a The user has further specified that he wants to scan forward to put Statement 1c at the top, and to open up one more level to view--it only remains for him to strike the "command accept" key.

6 Figure C-2(d) shows a subsequent view, after the user has scanned forward to put Statement 1c4 to the top, and has set the viewing parameters to all-all.

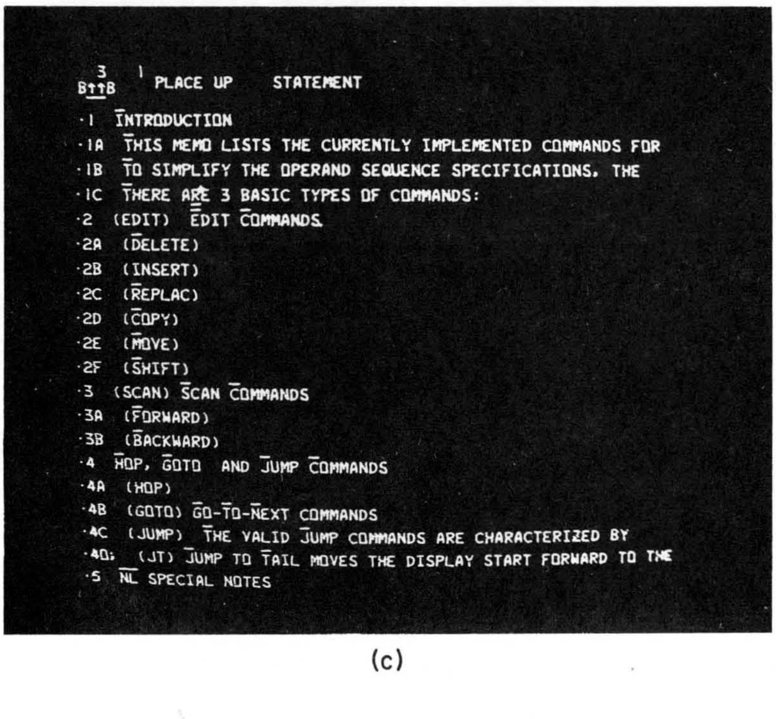
6a The last three characters of the echo register show that he has (by striking the letter X) designated a "Break Statement" command, made a selection of the word after which he wants the statement break to occur (the underline shows that he has selected the word "commands"), and that he has designated (by the "d") that he wants the newly broken-off statement to be one level below its "parent."

6b At the right of the command feedback line, the computer displays the location number that it will thus assign to the new statement. This feedback follows immediately upon the "u" or "d" designations of the user, and he may run this number up or down its levels as often as he wants by a succession of such characters, before he hits the "command accept" key to execute the command.

6b1 Further "d" entries from the illustrated state would have no effect, since a statement immediately following a third level statement (such as Statement 1c4) may not logically be of a level greater than 4.



(b)



(c)

FIG. C-2 Continued

6b2 The possible location numbers which this newly broken-off statement could be given would be 1c5 (the one offered first by the computer after selecting the breaking point), 1c4a, 1d, or 2.

6b3 If two "u" characters had been entered instead of the "d" before this command was executed, the new statement would be given a location number "2," and what are now Statements 2 and 2a would be automatically changed to Statements 3 and 3a.

7 Figure C-2(e) shows an example of a "frozen statement."

7a The sequence of statements viewed on the screen is not in natural order--Statement 2 would not logically immediately preceed the other displayed statements.

7b Statement 2, in fact, is "frozen" at the top of the screen, and this part of the screen remains unchanged as the user scans to different views within the file by any of the normal means of getting about. It is also unaffected by associated normal changes of viewing parameters.

7c The section of the screen displaying Statements 2b4 and below show that the user has moved his viewing position to Statement 2b4, where he has his viewing parameters set to all-all.

7c1 While holding the frozen statement on the screen in this fashion, material may be copied or moved between it and the rest of the screen in a normal fashion.

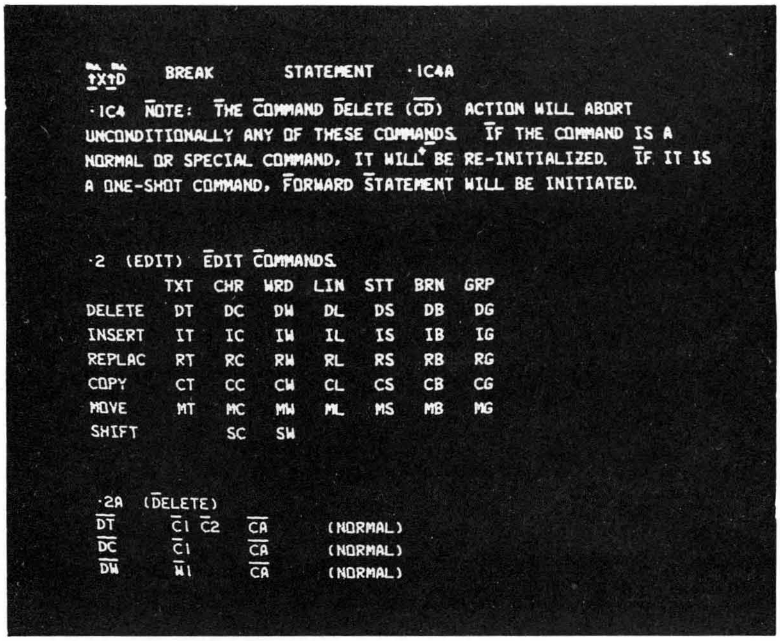
7d From the command feedback line, it is apparent that the user has arrived at Statement 2b4 by using the "Hop-to-Name" command, and that he designated the name of Statement 2b4 (i.e., "dl") by a select action on the word "dl" on the frozen statement.

7d1 It is evident from the echo register that this is at least the second successive such Hop-to-Name action that he has made--it requires two of the underlined UPARROW characters to do the select-execute sequence for each such hop action.

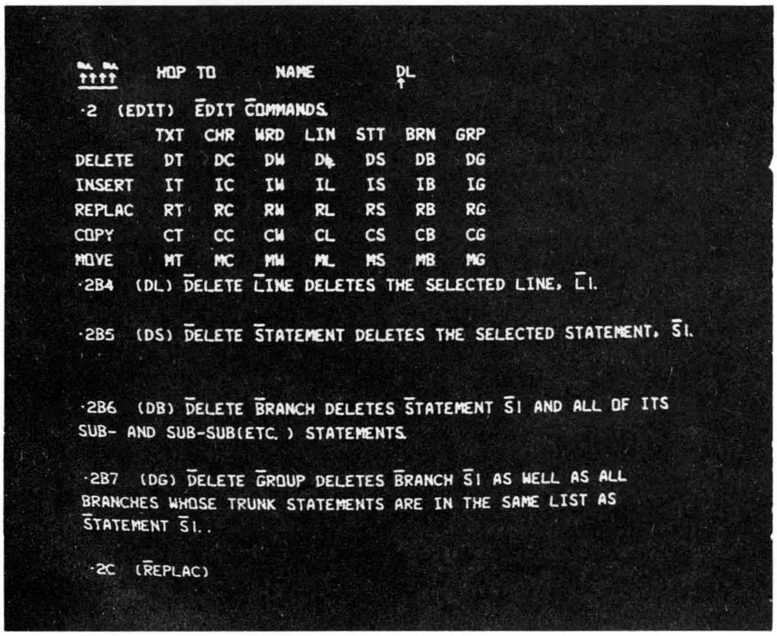
7e Statement 2 represents a table of editing commands, with each of the five basic editing operations being applicable to any of the seven types of text entities.

7e1 The two-letter entities in this table represent the mnemonic character pairs which may be used to specify the corresponding command (e.g., "dl" will specify the command "delete line").

7f By naming each statement wherein he has described the nature of a given command with the character pair used to designate that command,



(d)



(e)

FIG. C-2 Concluded

the user has cleverly harnessed this Hop-to-Name capability to allow him to use this table as a "pointable" index to all of the statements thus describing the various commands.

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 333 Ravenswood Avenue Menlo Park, California 94025	2a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED 2b. GROUP n/a

3. REPORT TITLE

STUDY FOR THE DEVELOPMENT OF HUMAN INTELLECT AUGMENTATION TECHNIQUES

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

Interim Progress Report

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

Douglas C. Englebart, William K. English, John F. Rulifson

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
March 1967	98	10

8a. CONTRACT OR GRANT NO.	9a. ORIGINATOR'S REPORT NUMBER(S)
NAS 1-5904 b. PROJECT NO. c. d.	Interim Progress Report, SRI Project 5890 9b. OTHER REPORT NO(S) *(Any other numbers that may be assigned this report)*

10. DISTRIBUTION STATEMENT

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY
	NASA-Langley Research Center, Mail Stop 126 Langley Station Langley, Virginia 23365

13. ABSTRACT

This report covers the first year of a continuing project, at the eighth year of a growing, multiproject program that is exploring the value of computer aids to augmenting human intellectual capability. Outlined briefly are the background and the "bootstrapping" nature of the program (e.g., the report was produced with the experimental computer aids) and its status at the start of this project. Advances during the year were in the programming system (CRT on-line debugging, etc.), in both off-line and on-line computer-aided text-manipulation systems implemented on a CDC 3100, and in specifications and designs for a time-shared system to be implemented in the next year on an SDS 940, which will be committed to serving the on-line CRT consoles for this program. Experiences with usage to date lead to an assortment of future research possibilities for both the computer aids and their utilization.

DD FORM 1 NOV 65 1473 (PAGE 1)

S/N 0101-807-6801

UNCLASSIFIED

Security Classification

14. KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
man-machine interface						
on-line						
real-time						
computer-aided text editing						
computer-aided text manipulation						