1972

RADC-TR-72-232
Final Report
8 June 1972

SRI Project 8457

ONLINE TEAM ENVIRONMENT

Network Information Center and Computer Augmented Team Interaction

Augmentation Research Center
STANFORD RESEARCH INSTITUTE
MENLO PARK, CA. 94025

Sponsored by
Defense Advanced Research Projects Agency
ARPA ORDER NO. 967

Rome Air Development Center
AIR FORCE SYSTEMS COMMAND
GRIFFISS AIR FORCE BASE, NEW YORK

RADC - TR - 72 - 232
Final Report
8 June 1972

SRI Project 8457

ONLINE TEAM ENVIRONMENT

Network Information Center and Computer Augmented Team Interaction

Augmentation Research Center
STANFORD RESEARCH INSTITUTE
MENLO PARK, CA. 94025

Sponsored by
Defense Advanced Research Projects Agency
ARPA ORDER NO. 967

Rome Air Development Center
AIR FORCE SYSTEMS COMMAND
GRIFFISS AIR FORCE BASE, NEW YORK

Do not return this copy. Retain or destroy.

ONLINE TEAM ENVIRONMENT

Network Information Center and Computer Augmented Team Interaction

Contractor: Stanford Research Institute
Contract Number: F30602-70-C-0219
Effective Date of Contract: 8 February 1970
Expiration Date of Contract and Amendments: 8 May 1972
Amount of Contract: $2,676,533
Program Code Number: 62706D

SRI Project Number: 8457

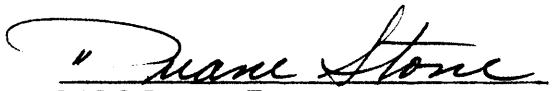
Principal Investigator: Douglas C. Engelbart
Phone: (415) 326-6200, ext. 2220

Project Engineer: Duane L. Stone
Phone: (315) 330-3857

This research was supported by the Defense Advanced Research Projects Agency of the Department of Defense and was monitored by D. L. Stone, RADC (ISIM), GAFB, NY 13440 under Contract F30602-70-C-0219.

PUBLICATION REVIEW

This technical report has been reviewed and is approved.



RADC Project Engineer

CREDIT

The research reported here is the product of conceptual, design, and development work by a large number of persons; the program has been active as a coordinated team effort since 1965.

The work from 8 February 1970 to 9 May 1972 involved the whole ARC staff:
Marilyn F. Auerbach, Walter L. Bass,
William S. Duvall, Douglas C. Engelbart,
William R. Ferguson, Beau A. Hardeman,
Martin E. Hardy, J. David Hopper,
Charles H. Irby, Mildred E. Jernigan,
Diane S. Kaye, Michael D. Kudlick,
Linda L. Lane, Harvey G. Lehtman,
Donald Limuti, Priscilla M. Lister,
N. Dean Meyer, Jeanne B. North,
James C. Norton, Cynthia Page,
Bruce L. Parsley, William H. Paxton,
Jeffrey C. Peters, Ralph Prather,
Jake Ratliff, Paul Rech, Barbara E. Row,
Jacques F. Vallee, Edwin K. Van De Riet,
Dirk H. van Nouhuys, Kenneth E. Victor,
Donald C. Wallace, Richard W. Watson,
and James E. White

In addition, the following consultants:
Don I. Andrews,
James A. Fadiman

and the following former members of the staff:
Geoffrey H. Ball, Roger D. Bates,
Vernon R. Baughman, Mary G. Caldwell,
Roberta A. Carillon, David G. Casseres,
Donald Cone, Mary S. Church, Robert L. Dendy,
William K. English, Ann R. Geoffrion,
Jared H. Harris, John T. Melvin,
Martha E. Trundy, and John M. Yarborough

ABSTRACT

1

TIME COVERED 1a

This report covers in detail work from February 1971 to May 1972. 1a1

TENEX 1b

During that time our PDP-10 and accompanying TENEX time-sharing systems became operational. We have made small adaptations in TENEX and developed a system that sends and retrieves files from tape archive. 1b1

NETWORK INFORMATION CENTER 1c

Use of the Network Information Center has increased steadily, including regular creation by experimenters at several sites of special-purpose documents on our system and severalfold increase in documents stored and cataloged, both online and in hard copy dispersed at the sites. We have prepared and dispersed manuals and given regular courses in our system to classes gathered from the Net. In the last weeks of the contract our display system ran experimentally from another site for the first time. 1c1

HARDWARE 1d

We have added a Bryant Drum, Digital Equipment Corporation RP02 disc packs, and leased more 30-character-per-second thermal printing terminals and compatible cassette recorders. 1d1

NEW FEATURES IN NLS 1e

To our online system we have added: 1e1

a command language, DEX, which allows entry of text on tape for later automatic processing into NLS files; 1e1a

several features which allow users to draw on the power of NLS more effectively, including individual control of a buffer for compiling various special purpose programs; 1e1b

cross file editing to our typewriter-oriented command language, TNLS, along with other features that suit NLS to typewriter terminal work; and 1e1c

to our display system, the capacity to split the screen, load several files at once, and transfer information from one file to another. 1e1d

We have begun the redesign of NLS in modular

units which will, among other things, ease transfer of all or part of NLS to other systems. 1e2

MANAGEMENT SYSTEMS 1f

In management applications we developed a first cut task-and-assignment management record-keeping system, made ever-growing use of our dialog support system in management, and, near the end of the contract period, reorganized our group into operational and project subgroups (a matrix organization) with projects oriented toward needs outside ARC. 1f1

CONTENTS

SUMMARY

3

INTRODUCTION

3a

We are developing a system of online computer aids for augmenting the performance of individuals and teams engaged in intellectual work and an Information Center for the ARPA Computer Network. This document reports hardware and software development and applications in several areas, and summarizes plans for continuing development. 3a1

We discuss here the work performed under a contract which extended from February 8, 1970 to May 9, 1972, but recount in detail only work of the last fifteen months of that period. Our work from February 8, 1970 until February 8, 1971 is reported in the Interim Technical Report, dated 30 June 1971, NETWORK INFORMATION CENTER AND COMPUTER AUGMENTED TEAM INTERACTION, RADC-TR-71-175, AD 737 131 (8277,). The 1970 work is summarized below but discussed in the body of this report only where necessary to explain developments of the last fifteen months. 3a2

To take advantage of the automatic reference search of our online system, bibliographic citations in this report are a little unusual looking. They will appear in two forms: 3a3

"See-- + a comma + a string of numbers and letters + a right parenthesis" *(e.g., See--,9an)* cites some other part of this report as identified by the statement numbers printed right. Online, a reader may cite such an address and move automatically to the appropriate part of the report. 3a3a

A four- or five-digit number in parenthesis *(e.g. (8277,))* cites a document in ARC's collection. The number is the ARC catalog number. Most of the documents cited in this report are online and an online reader may move to that file automatically as above. A reference section at the end of each chapter supplies bibliographic information about these documents in the usual way. 3a3b

A glossary appears in (,9). 3a3c

ARC has begun to maintain online a detailed description of the current state of its activities, the Handbook discussed below (see --,4b). 3a4

The reader may find in documents cited in the Handbook more detailed accounts of several matters than appear in this report, particularly of command and computer languages. 3a4a

The detailed accounts are in the following Handbook documents: 3a4b

TREE META (10869,), 3a4b1

DEX USER GUIDE (9934,), 3a4b2

DNLS PRELIMINARY USER GUIDE (10703,), 3a4b3

NIC TNLS USER GUIDE (7470,), 3a4b4

NIC JOURNAL USER GUIDE (7635,) 3a4b5

L10 PROGRAMMING GUIDE (USER GUIDE) (9246,) 3a4b6

L10 - A Programming Language for the Augmentation Research Center (Systems Programmer's Guide) (7052,) 3a4b7

HIGHLIGHTS OF 1970 3b

During that year we devoted our attention especially to our continuing effort to improve the efficiency of our online system and broaden and strengthen its usefulness to systems programming, to working with the ARPA Network, and to augmentation of distributed teams. 3b1

During the latter part of the year we were deeply involved with translating our software into forms compatible with a PDP-10 and with choosing and connecting its peripheral equipment. 3b2

We planned and began use of an important new group of tools for users which we now call User Programming. They are routines in which the basic user features of our online system are building blocks in construction of programs that carry out specific, rather complicated tasks, such as changing the order of a citation index and at the same time the format of the citations. Important User Programs are the rewritten Content Analyzer, the Analyzer Formatter, the Collector Sorter, and Executable Text. 3b3

Early in 1970 we developed an arithmetic and algebraic calculator package to our online system. The calculator has not yet been transferred to the PDP-10 version of NLS. 3b4

1970 saw new concentration on augmenting teams performing work that is distributed in time, space, and discipline. By way of communication and archival and managerial record keeping, we added a mail system and a Journal system. Any user might write a mail message from his terminal to any other users. The message was automatically brought to the recipient's attention when he logged in. Mail was particularly useful to our people temporarily or permanently at a distance from the Center. Mail messages automatically became part of the Journal. 3b5

The Journal is an online repository of the thoughts, records, baselines, and evolving designs of the group. Online is an index to the complete journal, including various retrieving aids such as sorting by title words. 3b6

Our participation in the ARPA Network in 1970 included: using University of Utah's PDP-10 via the Network to aid in our transfer to a new PDP-10, and development of the Network Information Center (NIC). 3b7

> In using the Net to re-program our PDP-10 we typically sent blocks to UXAH that consisted of relocatable binary data produced by compilers executing in our XDS-940 and producing code for the 10. The data was stored on a disc at Utah by the network control program so that someone here could reconnect and call on the Utah loader for the transmitted file. We found this service so useful that we added multiplexing at this end so that three of our programmers could use the Utah system at once. The link to Utah operated daily from August 1970 through January 1971 and constituted the most substantial data transmission over the Net to that date. 3b7a

In 1970 we established a collection of documents that form the basis of the Network Information Center, established online techniques for handling the documents, and, most important, began working dialog with the other centers. The combination of our reference data storage techniques with our programming allows retrieving documents according to a variety of attributes and combinations thereof; e.g., year of publication combined with author, or sponsoring

institution. We organized with the other sites on the Network to establish Station Agents to handle their interaction with the Network Information Center and supplied the Station Agents with a catalog of their collection and other working materials. To stimulate dialog, pending full operation by connected computers, we set up a central telephone exchange and a system for circulating documents and memos by U.S. Mail through the NIC, including an intra-net document numbering system. 3b8

In the Spring of 1970 we decided that DEC's PDP-10 with associated software and paging box from BBN might be a way to increase the number of consoles and displays available to us, to strengthen our system in other ways, and to ensure a system that could be expanded further with ease. In June after investigating several competing machines, we ordered a PDP-10 which was delivered in September. Our 940 was removed February 1, 1971. Associated equipment for the PDP-10 includes 128K of 1.0-microsecond core and the BBN Paging Box. After studying the various alternatives, we retained from the 940 system a 32K-word Ampex external core, UNIVAC drums as a swapping device, and a Bryant Disc for mass storage. A drum/disc interface, an interface for the external core system, and an I/O control box were built locally to our specifications. 3b9

Re-programming for the PDP-10 created the necessity and opportunity for thorough-going revision of our software. Our online system which had been written in a special language, SPL, was rewritten in L10, a language much more machine independent and more flexible in application. Our NLS was rationalized to allow more routines to call on other routines. Display routines were changed to allow division into up to eight areas which the user can load and edit independently. Many other features such as Mail, Journal, calculator were substantially improved in the transfer. 3b10

HIGHLIGHTS OF 1971 3c

Team Augmentation 3c1

In the last 15 months our work toward Team Augmentation has fallen into five areas: improvement of our dialog support system, the initial work on our handbook, our baseline record system, development of basic NLS, and reorganization of our laboratory staff. 3c1a

Dialog Support System 3c1b

As with the XDS-940 Journal system, the PDP-10 Journal system serves as an open-ended information storage and retrieval system, oriented toward recording the thoughts, notes, designs, workpieces, and reports communicated by users. 3c1b1

ARC and Network personnel use the Journal system daily. 3c1b1a

Since it became operational in April, 1971, approximately 1600 documents have been generated at ARC and submitted to the Journal. 3c1b1a1

The PDP-10 Journal system provides for automated entry of online documents in contrast to the essentially manual technique used on the XDS-940. 3c1b2

When a user submits a document, the system tags it with a number and a distribution note which later directs delivery of the document to a list of recipients the user spells out. 3c1b2a

A read-only copy of the submitted document is then stored, along with information relevant to the submission of the document (date/time, title, keywords, etc.). 3c1b2a1

A background process will subsequently transform this into the final and permanent Journal entry. 3c1b2a2

Delivery of Journal submissions to authors and recipients has been automated on the PDP-10 System. 3c1b3

Hard copy is automatically formatted and printed with an address page so that mailing simply involves folding, stapling, and stamping. 3c1b3a

An online delivery technique has been developed wherein a user may receive notice of documents addressed to him by the placement of statements in his initial file. 3c1b3b

These statements contain a link to the document, along with the sender's identification,

date/time of submission, document number, and title. 3c1b3b1

A message facility has been incorporated in the PDP-10 Journal, which eliminates the mail system used on the XDS-940. 3c1b4

Online Journal documents may now be reached through NLS by simply using the Catalog number as a file name. 3c1b5

The improved access to Journal documents has resulted in increased linking between Journal documents, whereby dialogs may involve a number of documents, all interlinked. 3c1b5a

Handbook 3c1c

We have begun development of a "Handbook," a "super-document" that contains the beginnings of an up-to-date, large, detailed, highly cross-referenced and well-indexed description of ARC project-team activity. 3c1c1

Such a document will provide ARC, as a team tackling complex system-development projects, with the highest-possible visibility over its working environment. 3c1c1a

Toward the end of the contract period we set up a team to design a Handbook system which will be used to construct, index, and maintain this document. 3c1c1b

Baseline Record System 3c1d

We constantly face more opportunities for changes or additions to our evolving system than we have resources to carry out. Therefore we have attempted to use NLS to find ways to make ever more effective, coordinated analysis of our ideas, and of our people, system, and material resources. 3c1d1

The result of such coordinated analysis is the adoption of a current visible plan, or "baseline" of expected events, agreed upon system developments, their external configurations, and resource allocations. 3c1d2

The information relative to the planned system developments is contained in our Baseline Record. 3c1d2a

The Baseline Record is a special subcollection of the Journal. It consists of a series of files specially formatted to contain task and resource allocation information, including particularly files of plans, specifications, analyses, designs, etc. 3c1d3

The present Baseline Record system has concentrated on the recording of information relevant xo individual tasks being performed or under consideraton by various ARC staff members. 3c1d4

There now are over 200 tasks of various magnitudes to consider in our planning and operational environment at any point in time. These range from simple bug-fixing to complex design or implementation tasks that may be performed by several people over many months. 3c1d4a

We have developed a set of programs with an initial data storage system that organizes information recorded about these tasks with features that permit routine summary views to be produced and that also make available flexible, user-created views of the Baseline task information. 3c1d4b

Procedures have been developed for data collection and input and for view production that aid in weekly updating of the Record. These views are produced in hardcopy and are also entered into the Journal. 3c1d4c

We are not satified with the present Baseline Record system. 3c1d5

We feel that our ARC users were not well guided and trained in BRS use and 3c1d5a

the initial system did not produce views that were useful enough - mainly because most of tne needed data were not in the system. 3c1d5b

Although we have started using ARC's Baseline Record system on a current task-by-task basis during tne past

year, we still need to develop a more complete, "higher level" picture of what new ARC system developments (functions, features, stages...) we want and expect to see. Among other considerations, this includes better definition of activity goals. 3c1d6

Basic NLS 3c1e

In this past contract period, we have taken several steps to further augment the software engineer 3c1e1

-- in fact, we have coined the acronym SEAS (for Software Engineer Augmentation System) to give specific system orientation towards the end of developing a full and balanced set of tools, techniques, methods, principles, etc. for augmenting software engineers. 3c1e1a

The developments described below are part of an accelerating activity -- an important part of our near-future plans in the next contract period involve a greater level of activity here. 3c1e1b

TNLS and DEX 3c1e2

A new and effective typewriter version (TNLS) has found wide use both at ARC and at sites on the ARPA Network. 3c1e2a

Improvements have been made in the display version (DNLS), 3c1e2b

and a first version of an offline mode (DEX) has been introduced. 3c1e2c

Changes that make possible cross-file editing allow any two passages to be involved by a given command. 3c1e3

In TNLS, addresses in a command may be "links" which can call any passage in any file on the system; 3c1e3a

in DNLS, split screens allow the user to view any two passages and control cross-file editing visually. 3c1e3b

Viewspecs make possible selective assimilation of information from one file into another. 3c1e3c

New special purpose subsystems have been developed or improved. 3c1e4

These include a sort-merge system, a user program system, and the output processor. 3c1e4a

Language development has continued. 3c1e5

At present the primary language systems developed and in use at ARC are the Tree-Meta Compiler-compiler System and the L10 Programming language system which was written in Tree-Meta. 3c1e5a

Work is currently progressing on a Modular Programming System (MPS) in collaboration with a group at the Xerox Palo Alto Research Center. 3c1e5b

Internal Organization 3c1f

During the past year, several ARC organizational arrangements were introduced, centering, in the early part of the period, mainly on line-activity structure and associated roles. 3c1f1

The creation of pusher (task leader) roles for tasks and coordination roles for system architecture, methodology, and personnel resources placed the responsibility more directly on selected individuals. 3c1f1a

Pusher roles were defined in the framework of the developing Baseline management system. Coordinating roles were also carried out in this environment. Our techniques for performing these roles still leave much to be desired. The planned recording of task requirements and designs in the journal will strengthen the roles. 3c1f1a1

In the Fall of 1971, we set up a four-man Executive Management Committee (EMC) to carry out much of the day-to-day operating management. 3c1f1b

SRI-ARC 8 JUNE 1972 13041
Summary
Highlights of 1971

During the past few months Dr. Engelbart has established, a new, broader overall organizational structure. 3c1f1c

This structure consists of three main activities that cover our framework and goal setting, line operation, and personal and organizational development needs. 3c1f1c1

These activities are called: FRAMAC, LINAC, and PODAC. 3c1f1c2

FRAMAC is to discuss and define the ARC intellectual framework and set longer-range goals and plans. 3c1f1c2a

LINAC is to carry out activities within the framework that move us toward the goals, including more detailed, shorter-range planning. 3c1f1c2b

PODAC institutionalizes continuing personal and organizational development. 3c1f1c2c

Network Information Center: Operations and Development 3c2

The ARPANET can be viewed as a collection of resources, people, hardware, software, data, and special services which can be brought together for short or long periods to work cooperatively. 3c2a

Built upon hardware and fundamental software connections are the processes that assist users to find the geographically distributed facilities they need to solve or study problems and to allow scattered people to work together effectively in tasks of mutual interest. 3c2a1

We see the Network Information Center (NIC) as one part of the ARPANET experiment that is interested in the latter problems. 3c2a2

The NIC helps to create and sustain the sense of community needed in an experiment such as that of the ARPANET. 3c2a2a

The NIC is not a classical information center because

it provides a wider range than bibliographic and library services. 3c2a3

The NIC Public 3c2b

One of the problems in the design of an information service is to determine the clientele and its needs. 3c2b1

Our initial analysis showed us four main needs: 3c2b2

Reference and General Network Information; 3c2b2a

Collaboration Support; 3c2b2b

Document Handling and Creation; and 3c2b2c

Training. 3c2b2d

The clientele for NIC appeared initially to be people developing and building the Network, who were to be followed by those whose research or development interests would be intimately connected with Network resources or who would be experimental users of various Network resources. 3c2b3

NIC Services 3c2c

To meet the above goals, the NIC services available at the end of the report period, May, 1972, through the Net were: 3c2c1

Online: 3c2c1a

(1) Access to the typewriter version (TNLS) of the Augmentation Research Center Online System (NLS) for communique creation, access, and other, experimental use. 3c2c1a1

(2) Access to Journal, Number, and Identification Systems which allow messages and documents to be transmitted to Network participants. 3c2c1a2

(3) Access to a number of online information bases through a special Locator file using NLS link mechanisms. 3c2c1a3

Offline: 3c2c1b

(1) A Network Information Center Station set up at each site with: 3c2c1b1

(a) A Station Agent to aid in use of the NIC. 3c2c1b1a

(b) A Liaison to provide technical information about his site. 3c2c1b1b

(c) A Station Collection containing a subcollection of documents of interest to Network participants. 3c2c1b1c

(2) Techniques for gathering, producing and maintaining NIC Functional Documents, such as: 3c2c1b2

(a) Current Catalog of the NIC Collection. 3c2c1b2a

(b) ARPA Network Resource Notebook. 3c2c1b2b

(c) Directory of Network Participants. 3c2c1b2c

(d) NIC User Guide. 3c2c1b2d

(3) General Network referral and handling of document requests. 3c2c1b3

(4) Building of a collection of documents potentially valuable to the Network Community. 3c2c1b4

In the beginning we've tried to collect documents valuable to network builders. 3c2c1b4a

(5) Crude selective distribution to Station Collections. 3c2c1b5

(6) Training in use of NIC services and facilities. 3c2c1b6

NIC Goals 3c2d

In the course of its evolution, the ARPANET will

continue to generate needs for new software services in interactive data management. 3c2d1

we propose to develop a user-oriented information facility based upon the NLS system and initially serving the needs identified in (,3c2a1). This information facility is a new step in the "bootstrapping" of the Augmentation Research Center, and is leading to the establishment of a new resource to be made available to ARPANET users. 3c2d2

Network Participation 3c3

Our Network participation outside of NIC activity has been in two main areas, protocol development through work in several protocol design communities and general Network coordination through membership on the short-lived Network Working Group Steering Committee and its successor, Network Facilitators Group. 3c3a

Computer Facility 3c4

Hardware 3c4a

At the end of the first year of this contract, we transferred our computer operations from an XDS-940 to a PDP-10 computer. The transfer effort is described in our interim report for the first year (8277,). 3c4a1

Hardware activity during the past year has focused on additional tuning of the new configuration, maintenance, troubleshooting and operation of the facility, and some upgrading of critical parts of the system. 3c4a2

Our hardware configuration contained a number of old, one-of-a-kind pieces of equipment brought over to the PDP-10 system from the previous XDS-940 system. These pieces of equipment have proven difficult to maintain and studies were launched on how to replace or upgrade this equipment. A new BBN network interface and a new DEC RP-02 disc system were installed in the spring of 1972, replacing older unreliable equipment. Hardware upgrading of our display system and its special core box has begun to provide temporary relief until a replacement system can be planned. An additional 32k

of core is to be added shortly. Studies leading to recommendations to add another channel, disc controller and set of disc drives have been completed. These additions will provide more file space and backup swapping capability. Improved reliability should begin to be manifest in the summer of 1972. 3c4a3

System Software 3c4b

TENEX 3c4b1

We cooperate actively with BBN and other users in debugging and maintaining TENEX, and have developed a few new features, both visible to users and internal to the system. 3c4b1a

Within the system: 3c4b1b

We have forsaken TENDMP for loading the monitor from DECTAPE and use instead DTBOOT from DEC. 3c4b1b1

We have added a jsys, a jump to a monitor subroutine, to say that padding (sending rubouts) is required for fast terminals when a CR or LF is output. 3c4b1b2

We have made many changes to the teletype routines to accommodate our displays. 3c4b1b3

To greatly simplify startup we have changed the starting address of the monitor from 100 (which goes immediately to DDT) to SYSG01. 3c4b1b4

We no longer add code to existing files when we get new monitor releases. Instead we have defined additional files that are assembled with each group of files and, where possible, have made our additions in these new files with JRSTs and CALLs to the new code. 3c4b1b5

We have modified the system such that if CHECKDSK does not run successfully, then nothing else, e.g. AUTO-STARTUP jobs, can run (except for the operator's console and one special dial-up line) until the disc has been fixed and CHECKDSK has run successfully. 3c4b1b6

(1) an increasingly useful service to the Network Community and 3c5a2a

(2) an important part of the Network experiment (in its distributed, collaborative operations and in its Network-utility role). 3c5a2b

And moving useful augumentation techniques and services out into the ARPA Network Community. 3c5a3

A central point of our proposed approach is our need to learn to negotiate and provide extensive services to distributed users. 3c5b

Therefore, we plan to concentrate our efforts within a four-pronged project wherein coordinated advances can be made: 3c5c

(1) Developing service functions that will be the most help to our above-mentioned goal structure, 3c5c1

(2) Developing the knowhow and capability for delivering significantly useful service to the Network, as a utility, 3c5c2

(3) Developing the knowhow and capability for marketing a utility service to the Network, 3c5c3

and wherein we become ever better at 3c5d

(4) Operating a utility service. 3c5d1

Depending on funding availability and other arrangements to be negotiated we may find ways to provide additional service capacity through placement of the computer-based portion of our augmentation system on a computer or computers operated for us by a commercial timesharing utility. 3c5d1a

REFERENCES 3d

(7052,) W. H. Paxton (SRI-ARC). L-10 - A Programming Language for the Augmentation Research Center (a systems programmer's guide). 29 May 1971. 46p. 3d1

(7470,) Network Information Center, Augmentation Research Center, Stanford Research Institute, Menlo Park, California 94025. TNLS User Guide: Preface, Syntax and Contents. 1 September 1971. Separately paged. 3d2

(7635,) Network Information Center, Augmentation Research Center, Stanford Research Institute, Menlo Park, California 94025. NIC Journal User Guide. 1 October 1971. Separately paged. 3d3

(8277,) D. C. Engelbart (SRI-ARC). Network Information Center and Computer Augmented Team Interaction, Interim Technical Report. Augmentation Research Center, Stanford Research Institute, Menlo Park, California 94025. Rome Air Development Center, ARPA. RADC-TR-71-175, AD 737 131. 30 June 1971. 104p. 3d4

(9246,) Augmentation Research Center, Stanford Research Institute, Menlo Park, California 94025. L-10 Programming Guide (a user guide). 4 April 1972. 100p. 3d5

(9934,) Augmentation Research Center, Stanford Research Institute, Menlo Park, California 94025. Deferred Execution (DEX) User Guide. 16 June 1972. 66p. 3d6

D. C. Engelbart (SRI-ARC). Experimental Development of a Small Computer-Augmented Information System. Annual Report Covering the Period 15 April 1971 through 15 April 1972. 22p. 3d7

(10703,) Augmentation Research Center, Stanford Research Institute, Menlo Park, California 94025. DNLS Preliminary Reference Guide. 21 June 1972. Separately paged. 3d8

(10869,) D. I. Andrews, H. G. Lehtman, W. H. Paxton (SRI-ARC). Tree Meta - A Metacompiler for the Augmentation Research Center. In process. Unpaged. 3d9

SRI-ARC 8 JUNE 1972 13041
Team Augmentation
The Journal

TEAM AUGMENTATION

by Charles H Irby, William H Paxton,
William S Duvall, James C Norton,
Bruce L Parsley, Mary S Church, Harvey G Lehtman,
Walter L Bass, J David Hopper, Douglas C Engelbart,
L Peter Deutsch, and James G Mitchell. 4

JOURNAL 4a

Introduction 4a1

As ARC becomes more and more involved in the augmentation of teams, we are giving serious consideration to improving intrateam communication with whatever mixture of tools, conventions, and procedures will help. 4a1a

If a team is solving a problem that extends over a considerable time, the members will begin to need help remembering some of the important communications--i.e., some recording and recalling processes must be invoked, and these processes become candidates for augmentation. To consider some of the different conditions where such storage and recall may be useful, suppose Person A communicates with Person B about Item N at Time T. 4a1b

They may well remember their exchange during the problem-solving period. But consider the case of person C who, it will turn out, is going to need to know about this communication at time TT: 4a1b1

Perhaps he was there at Time T, but 4a1b1a

he was too heavily involved even to notice the communication, and/or Item N was not relevant to his work at that moment and so was not implanted for ready recall. 4a1b1a1

Perhaps A and B did not anticipate his later need and thus failed to invite him into their interchange or inform him of its conclusion. 4a1b1b

Perhaps, although Persons A and B knew he would later need the information, they didn't want to interrupt their own working sequence with the

procedure of interrupting Person C and getting him involved. 4a1b1c

or, if the consequences of the interchange carry over into a long-lasting series of other decisions, one or both parties may fail to remember accurately, or may remember differently because of different viewpoints, and troublesome conflicts and waste of effort may result. 4a1b2

A single person will make a list of things to do on a shopping trip because he has learned that the confusion and pressure may make him forget something important. It is obvious that to be procurer for one of a mutually developed, interdependent pair of lists would make it even more important to use a record. 4a1b3

Further consider the effect if the complexity of the team's problem relative to human working capacity requires partitioning of the problem into many parts where each part is independently attacked, but where among the parts there is considerable interdependence through interactions on mutual factors such as total resource, timing, weight, physical space, and functional meshing. 4a1c

Here, the communication between Persons A and B may well be too complex for their own accurate recall. For example, their communication period resulted in scratch paper or a chalkboard covered with possibilities and the essence of the agreed-upon solution, which has since disappeared. 4a1c1

We envision augmenting our collaborative team by having a "Dialog Support System (DSS)," containing current and thoroughly used working records of the group's plans, designs, notes, etc. Therefore, we have begun to develop a system for entering and managing those records. The ARC Journal is the central feature of this intragroup documentation system. 4a1d

The DSS involves techniques for use by distributed parties to collaborate effectively by means of the inter-linked referencing between NLS files, particularly within the recorded-dialog medium of an NLS Journal. 4a1e

FIGURE 1. ARC Dialog Support System hardcopy cave. Third shelf from the top is the Handbook; fourth is the Baseline Records; fifth is Journal indices.

4a1e1

Our DSS will provide the following general online aids: multiwindowed displays; simultaneous and independent mobility and view control among many files; link-setup automation; back-link annunciators and jumping; aids for the formation, manipulation, and study of sets of arbitrary passages from among the dialog entries; integration of cross-reference information into hardcopy printouts. 4a1e2

It also will include people-system developments: conventions and working procedures for using these aids effectively in conducting collaborative dialog among various kinds of people, at various kinds of terminals, and under various conditions; working methodology for teams doing planning, design, implementation coordination, and so on. 4a1e3

The PDP-10 Journal 4a2

During 1971, implementation of the initial PDP-10 Journal system was completed. 4a2a

As with the XDS-940 Journal system, the PDP-10 Journal system provides us an open-ended information storage and retrieval, oriented toward recording the thoughts, notes, workpieces, and reports produced by users. 4a2b

The system is in daily use by ARC personnel. 4a2b1

Since the first version of the system became operational in April, 1971, approximately 1600 documents have been generated and submitted to the Journal. 4a2b1a

The system is also offered as a NIC service. 4a2b2

The PDP-10 Journal system provides for automated entry of Online documents in contrast to the essentially manual technique used on the XDS-940. 4a2c

An NLS user can submit any portion of an NLS file (which may or may not be currently in his viewing area) to the Journal without leaving NLS. 4a2c1

In order to do this, he simply executes a command which places NLS into a sub-command level which recognizes commands relevant to Journal operation. 4a2c1a

As a document is submitted, it is assigned a number, cataloged, and a distribution record is created which will later cause delivery of a copy of the document to a list of recipients indicated during the submission process. 4a2c2

A read-only copy of the submitted document is then stored, along with information relevant to the submission of the document (date/time, etc.) 4a2c2a

A background process will subsequently transform this into the final Journal entry. 4a2c2b

Delivery of Journal submissions to authors and recipients has been automated on the PDP-10 System. 4a2d

Hardcopy is automatically formatted and printed with an address page so that mailing simply involves folding, stapling, and stamping. 4a2d1

An online delivery technique has been developed wherein a user may receive notice of documents addressed to him by the placement of links in his initial file. 4a2d2

A message facility has been incorporated in the PDP-10 Journal, which replaces the mail system used on the XDS-940. 4a2e

Online Journal documents may now be accessed through NLS by simply using the catalog number as a file name. 4a2f

A catalog search is done which determines the real name and location of the file containing the document with the indicated number. 4a2f1

This search is transparent to the user, and once located, the document is loaded as if the user had typed in the name and directory information contained in the catalog. 4a2f2

User appearance 4a3

As the user initially addresses the Journal system for document submission, he must define the document as any legal NLS structural entity (Statement, Branch, Group, Plex, or File) or as a message (literal) to be typed in. 4a3a

The document is immediately assigned a catalog number, and copied into a work area. 4a3a1

As this is being done, information relevant to the document (date/time, author, etc.) is recorded in the document header, along with default parameter settings. 4a3a1a

The user is now placed into an interactive submode, where the following parameters relevant to document submission may be specified: 4a3b

Author: Person (persons) or group sponsoring the document. 4a3b1

Clerk: Person actually submitting the document. 4a3b2

comments: A comment which is kept in the document header as an appendage to the document. 4a3b3

Distribution: A list of persons or groups to receive copies of the document. 4a3b4

Keywords: Key words which may be used for document retrieval at a later time. 4a3b5

Obsoletes: A list of documents obsoleted by the document being submitted. 4a3b6

subcollections: A list of subcollections in which this document is to be included. 4a3b7

The subcollections listed here are in addition to: 4a3b7a

Any subcollections associated with the submitter by default. 4a3b7a1

Any groups included in the distribution list. 4a3b7a2

Title: A title for the document. This title will appear as a default page header in the final formatted version. 4a3b8

Updates: A list of documents updated by the document. 4a3b9

Additional to the parameter specification commands are: 4a3c

commands for control 4a3c1

Quit: Leave the Journal submission submode, and abort the entry. 4a3c1a

Go: Terminate the parameter specification phase and begin the actual document entry. 4a3c1b

status Command: Shows the current status of the entry parameters 4a3c2

place Link Command: Allows the user to specify a location in a file, which will be used for inserting a statement containing a link pointing to the submitted document when submission is complete. 4a3c3

Interrogate Command: Places the user in a passive rather than active interactive mode. Subsequent to this command, the system will request specification of certain parameters from the user. 4a3c4

After the user has initiated the Go command, the system proceeds to execute the necessary functions for making a Journal entry from the working document. 4a3d

When this process has been successfully completed, a link locating the just-submitted document is typed or displayed to the user. 4a3d1

The user is then returned to the NLS command mode. 4a3d2

The Journal System User Guide (7637,) provides additional information on the use of the system. 4a3e

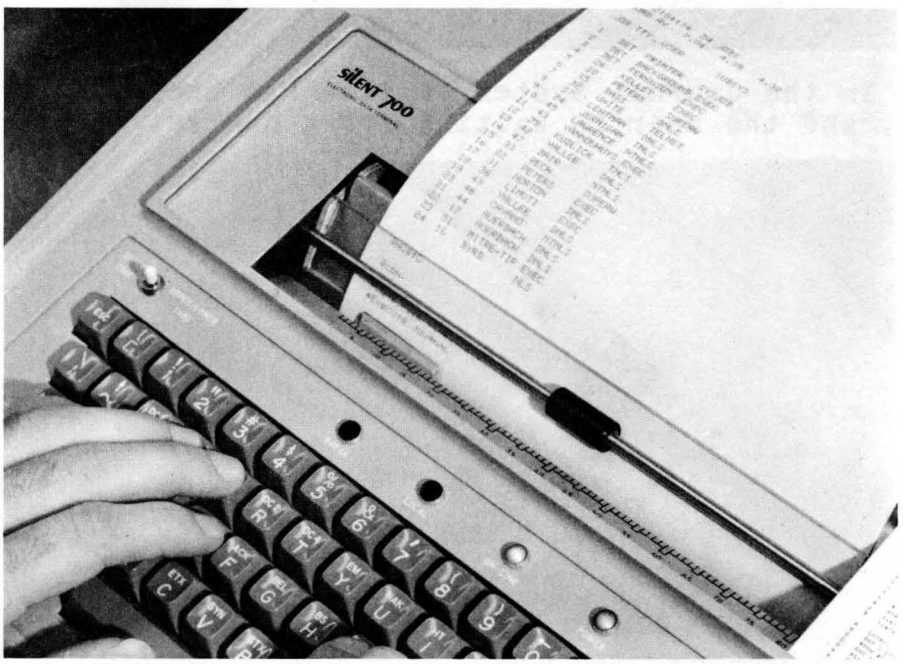


FIGURE 2. Someone calling the Journal System in TNLS.

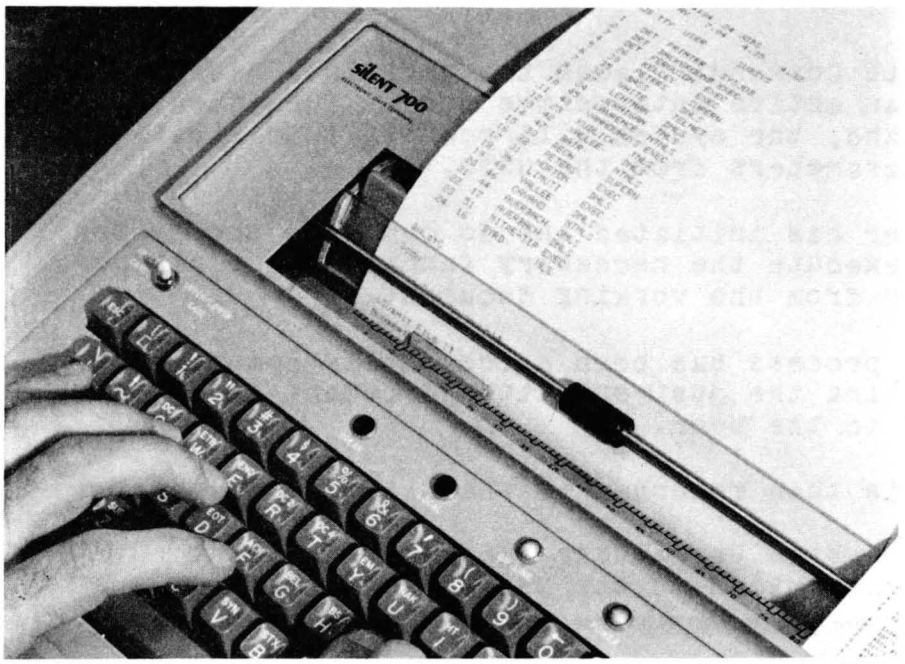


FIGURE 3. In the Journal System, a file has been submitted, and the user is waiting for a number.

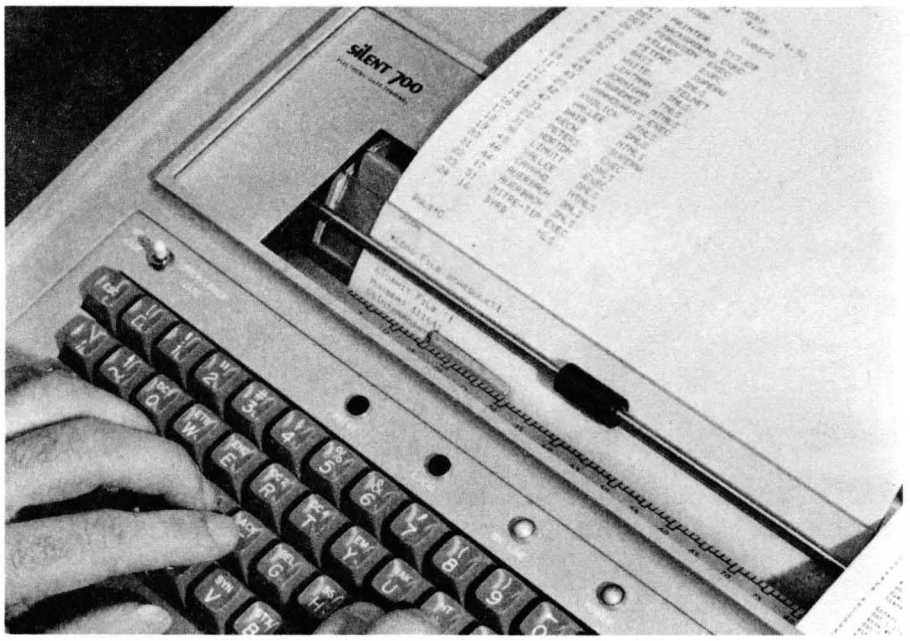


FIGURE 4. The user commands the System to quiz him for the information it needs.

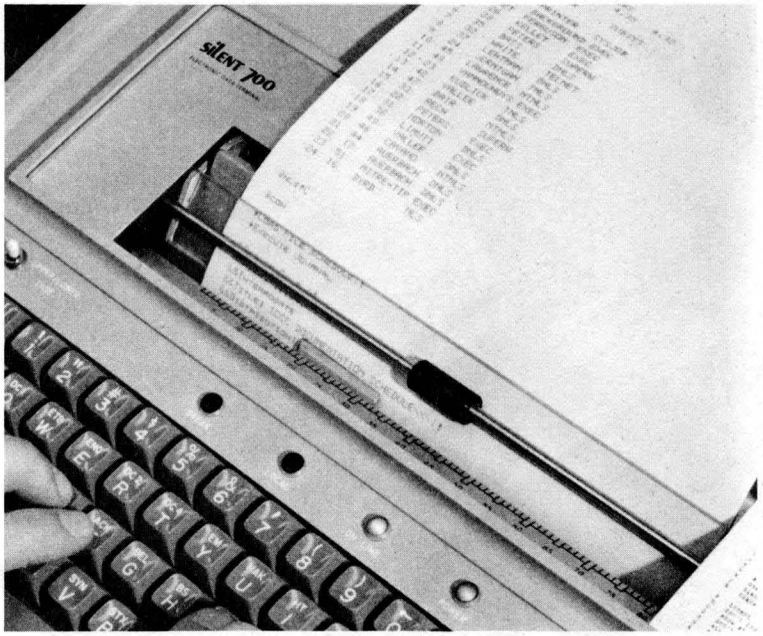


FIGURE 5. The user has responded to the System's promptings with the title and is about to fill in the distribution list.

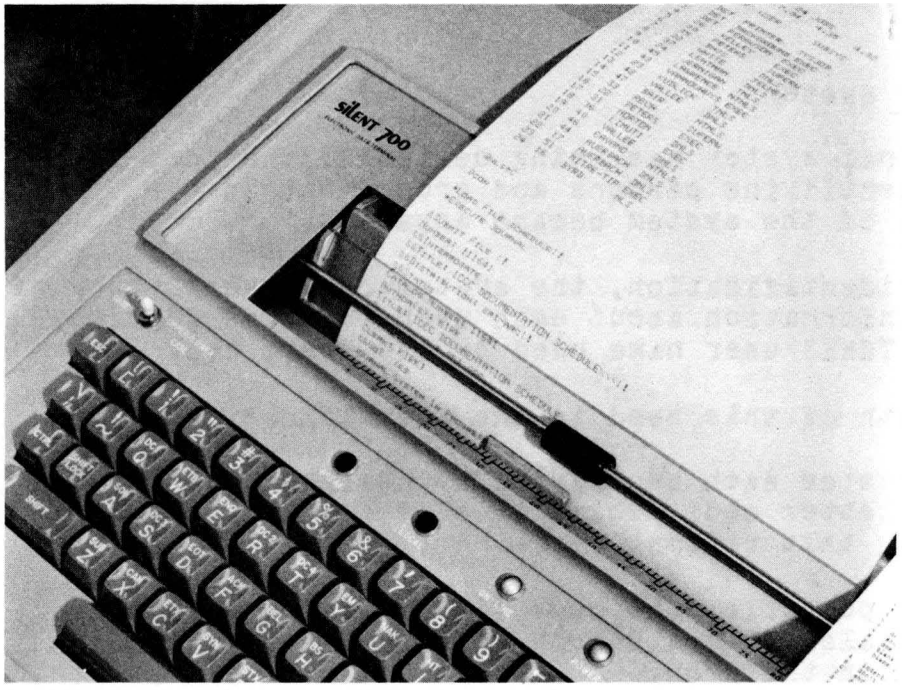


FIGURE 6. Journal System in progress.

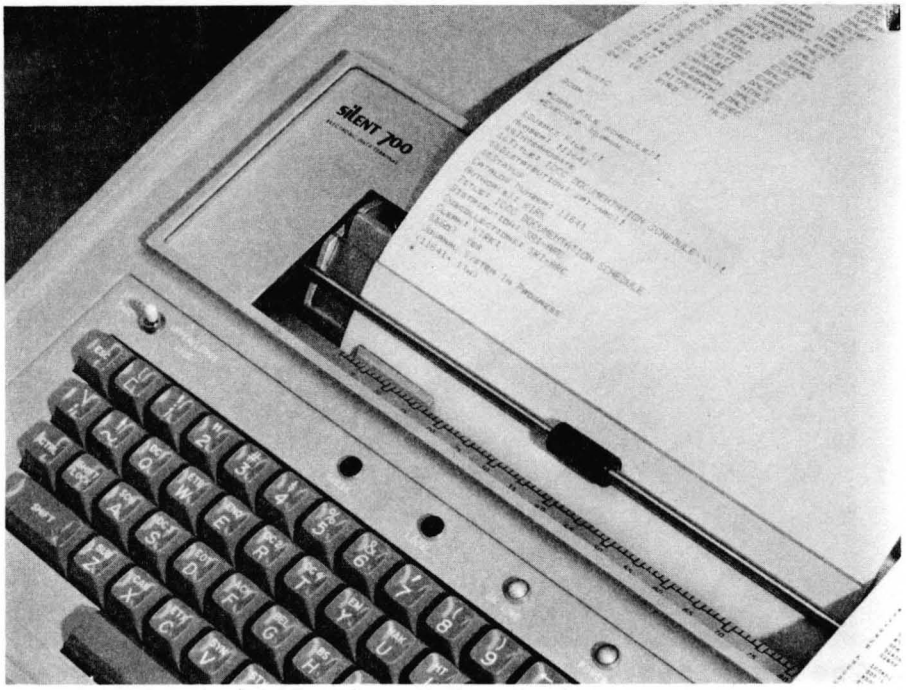


FIGURE 7. Submission complete.

Identification System 4a4

As the Journal system was being designed, the need for uniquely identifying persons and groups within the environment of the system became apparent. 4a4a

Given this identification, the system could keep track of a body of information about each user, such as address, telephone, TENEX user name used by the user, etc. 4a4b

The outgrowth of this need is the Identification system. 4a4c

With this system each user/group is assigned a unique two-to-six-letter code, which is subsequently used as a 'handle' for that person. 4a4d

Wherever possible, the code (IDENT) for a person is the initials of that person, and for groups the acronym for the group. 4a4d1

The IDENT may be used to locate an entry in a file which contains the necessary information about that person or group. 4a4e

Provided in the Identification system are not only handles for retrieving information about any IDENT, but a command sub-level for generating new IDENTS and modifying information for old ones. 4a4f

The Identification system is used extensively by all phases of the Journal. 4a4g

The Identification System User Guide (7638,) provides additional information on the use of the system. 4a4h

Number system 4a5

The Number system provides a capability for centrally assigning Master catalog and Network Working Group/Request for Comments (NWG/RFC) numbers. 4a5a

There is a set of NLS commands for directly assigning catalog numbers, and for pre-assigning RFC and Journal numbers. 4a5b

There is also a set of handles that allows numbers to be assigned to internal processes, e.g. the Journal. 4a5c

The Number System User Guide (7639,) provides additional information on the use of the system. 4a5d

Document Access 4a6

The XDS-940 Journal system provided essentially offline hardcopy access to Journal documents. 4a6a

With the PDP-10 Journal system, an effort has been made to provide convenient online access to Journal documents in addition to improved offline access. 4a6b

Hardcopy master and access collections (libraries) are maintained of all Journal documents. 4a6b1

While the master collection is maintained in its original form, documents from the access collection may be checked out, annotated, and copied by ARC personnel. 4a6b1a

The master catalog number is still the key to identifying documents. 4a6b2

As indispensable aids to the user, ARC provides author, number, and titleword indices. 4a6b3

These indices are automatically produced from the ARC Master catolog by a series of L10 user programs. 4a6b3a

ARC JOURNAL INDEX BY AUTHOR 4a6b4

Title	Date	Number	Author
* The TENEX Scheduler	21 Jul 71	7419	Andrews
* response memo	21 Jul 71	7415	Andrews
* response memo	12 Jul 71	7394	Andrews
* No title	14 May 71	6982	Andrews
* known performance problems	28 Feb 72	9313	ARC System Mea
* SCHED1 problem	18 Feb 72	9232	ARC System Mea
* Pod Reshuffle Proposal	13 Jul 72	11041	Auerbach
* Reply to 10954 (reply to 10953)	10 Jul 72	11002	Auerbach
* Reply to your comments on NIC documentation	5 Jul 72	10953	Auerbach
* re: update	5 Jun 72	10633	Auerbach
* let me tell you about the Handbook	5 Jun 72	10632	Auerbach
* re: attach/detach and at vs.after	23 May 72	10566	Auerbach
* re: output processor directives (again)	11 May 72	10417	Auerbach
* Header Positioning Options	11 May 72	10413	Auerbach
* re: <documentation>request	8 May 72	10384	Auerbach
* PRIMER DRAFT	4 May 72	10332	Auerbach

A Portion of the Journal Author Index.

Online access is provided to all documents added to the Journal collection since the PDP-10 Journal system became operational. 4a6b5

Any Journal document may be located by using the master catalog number as a file name.

Regardless of the location of the document, the system will find it and return it to the user as requested.

At the present time, all recent and most earlier key documents are kept online.

An archival system is currrently being implemented.

With this system, a request for a document which is not in direct access storage will result in a response of the form : "Document is in Secondary Storage--Retrieve ?".

An affirmative response will cause the system to direct an operator to mount an appropriate tape (or disc pack) and load the file to direct access storage.

An algorithm based on access activity and priority will be used for determining which documents will be kept permanently in direct access storage.

As with the hardcopy collections, author, number, and titleword indices are provided online as an aid to locating documents.

Additionally, a user may use any level of L10 user programs and Content analysis patterns to process the Journal catalog, thereby creating his own sub-collections using whatever selection criteria he chooses.

Document Distribution 4a7

Document distribution is more convenient not only for the user specifying the distribution of a document, but also for the operator producing hardcopy, and the recipient. 4a7a

A user submitting a document may specify recipients by simply entering an IDENT for said recipient as one of the parameters specified during submission. 4a7b

Since an IDENT may identify either an individual or a group, distribution to many persons/groups may be specified in a simple manner. 4a7b1

E.g. "Distribution: SRI-ARC" indicates that a copy of the document is to be distributed to each ARC person. 4a7b2

Copies of any document in the Journal collection may be distributed in a like manner using the Secondary Distribution command. 4a7c

A user may specify the manner in which Journal documents addressed to himself are to be distributed. 4a7d

current delivery options are hardcopy and online. 4a7d1

A user may specify either or both of these options. 4a7d2

Other options will be provided as they become necessary.

If hardcopy delivery is specified, the user will receive a hardcopy version of all documents addresssed to him via the U.S. mail. 4a7d3

If online delivery is specified, notification of a document addressed to the user is received via a branch in the users initial file. 4a7d4

Included in the notification are the document author, number, date, and title; any comments or notes associated with the document; and a link locating the document.

Physical distribution of Journal documents is automated to a high degree. 4a7e

online delivery is done by a background processor which is automatically started when TENEX is initiated. 4a7e1

The printing of hardcopy must be initiated by an operator, but then the system proceeds to produce correctly formatted and addressed hardcopy without operator intervention (except for paper handling, etc.). 4a7e2

A provision has been made for automatically starting hardcopy production, but is as yet inoperative because of certain system interface problems.

The printed hardcopy must be subsequently stapled stamped and mailed. 4a7e3

Special Features 4a8

Certain applications of the Journal system have required special handling. 4a8a

Most notable of these special applications has been the Network Working Group Request For Comments (NWG/RFC). 4a8a1

The Journal and Number systems have been modified so that they provide the necessary functions for producing RFC's within the context of the Journal. 4a8a2

This greatly facilitates the processing and distribution of these documents. 4a8a3

Problems and Comments 4a9

Reliability 4a9a

In terms of file handling, the Journal is a complex system. 4a9a1

One of the major problem areas has, correspondingly, been file manipulation, specifically file integrity. 4a9a2

There are (at least) 4 files which must contain synchronized data for each Journal entry. 4a9a3

Due to a variety of factors (such as disc errors and TENEX bugs) one or more of these files has occasionally been destroyed. 4a9a4

Unless the Journal system immediately recognized this fact, any subsequent Journal entries could potentially cause significant scrambling of related data, resulting in numbers being assigned twice, documents being delivered two or more times (or not at all), or documents disappearing.

Several efforts have been made to make the Journal fail-soft in this area. 4a9a5

Whenever the system is restarted, a special verification and repair program is automatically run.

This program checks the integrity of Journal files, and (if possible) fixes any errors it finds. If an error is found which cannot be automatically fixed, a message is typed on the operator and logging consoles, and the Journal system is locked.

Periodically, a background process runs and checks the validity of various files. Again, if any errors are found, the Journal is locked.

If any file errors are discovered during the submission process, the Journal is locked and any user currently in the process of submitting a Journal document is notified of a file error, and is returned to the NLS command level.

Operations 4a9b

Despite efforts to make the Journal fail-soft, an error occasionally occurs that is not immediately detected. 4a9b1

When this occurs, the result is frequently a mess that requires several hours of manual fixup to restore the Journal mechanisms to their proper state.

This creates an environment which makes reliable operation of the Journal system difficult and subject to the whims of a sometimes unmerciful system.

Fortunately, however, increased reliability of the system (due largely to the RP02 Disc Packs and improved techniques of maneuvering within the constraints of TENEX) has sharply decreased the frequency of serious file crashes.

The major current cause is running out of Disc space, which TENEX does not handle very gracefully.

For an extended period, there has been an interface problem between TENEX and the part of the system which produces hardcopy. 4a9b2

Again, this is in the area of file handling.

This asynchrony has made consistent production of hardcopy difficult. In fact, for a while it was virtually impossble.

The hardcopy production system will not be smooth and automatic until the interface problem is rectified, which will hopefully be the case in one of the (not too distant) future releases of TENEX.

Summary 4a10

The Journal system (along with the Identification and Number systems) is currently a viable system in use by ARC and Network personnel. 4a10a

There are certain efficiency problems, largely due to certain system file functions requiring greater overhead than originally anticipated, and our attempt to implement the Journal system using NLS files for the data base. 4a10b

Manipulation of NLS files is considerably slower than the manipulation of specially formatted files would be. 4a10b1

Future efforts will attempt to improve the efficiency. 4a10b2

Other systems and procedures within the ARC and Network environments are interfacing with the Journal system. 4a10c

The Baseline Record System uses the Journal system for the distribution of task lists and other planning information to ARC personnel. 4a10c1

The Journal will use a new Catalog Production System for the creation of its catalogs. 4a10c2

The Journal is an integral part of the ARC Handbook activity. 4a10c3

The Journal system is being actively used in design processes and dialog not only in ARC, but among Network users as well. 4a10c4

Future Journal system changes and additions will attempt to improve the handling of problem areas, as well as introducing new tools for viewing, retrieving, and linking among Journal dialogs. 4a10d

A major Dialog Support System effort will be in the creation of a set system, which will allow the flexible and convenient manipulation and viewing of collections of Journal items. 4a10e

HANDBOOK 4b

Description 4b1

The ARC Handbook is intended as a "super-document" containing an up-to-date, large, detailed, highly cross-referenced and well-indexed description of ARC project-team activity. 4b1a

Such a document will provide ARC, as a team tackling complex system-development projects, with the highest possible visibility over its working environment, i.e. over its: 4b1b

- planning -- plans, contingency alternatives, resource commitments, status, criticisms 4b1b1
- designing -- designs, design principles, constraints, estimates, analyses, supportive data, relevant needs and possibilities 4b1b2
- operating -- roles, task definitions, assignments, policies, operational procedures and conventions 4b1b3

ARC has formed a team whose responsibility is the design of a Handbook system which will be used to construct, index, and maintain this document. However, concurrent with a formalized Handbook design is a bootstrap attempt to pull together bits and pieces of ARC information from sources at hand. The latter is described here. 4b1c

At present, we have just finished the first and very primitive pass at organizing and obtaining in hardcopy much documentation relevant to the contents of an ARC Handbook. An outline is included in this report, see--,11). 4b1d

The Handbook is arranged topically; this arrangement is by no means fixed as we expect to learn much from actual usage and will redesign as appropriate. It exists online much as the Contents appear here but the online version includes links to each of the documents referenced. It also exists in hardcopy in the ARC library and includes a copy of each of the documents referenced. Procedures have been written which describe revision/maintenance for the Handbook in its current form. 4b1e

At this writing, the primary guide to the Handbook is the contents file reproduced in this report. A simple keyword index will be written in the near future and eventually, a system for automatically producing indexes. 4b1e1

The Handbook as it now exists is by no means inclusive as its primary source is the Journal for information about system features, ARC procedures, etc. However, the building of the Handbook has revealed and specified many areas of insufficient documentation and journalization and as such has already stimulated documentation and journalization activity at ARC. 4b1f

It is currently being used as an aid to some individuals and documentation teams in the production of general, medium-scale and medium-complexity documents. This usage is expected to increase as people become more familiar with its organization, reliability, and inclusiveness. 4b1g

BASELINE RECORD SYSTEM 4c

Introduction 4c1

Our ARC system development team has the same basic needs for planning, coordinating, documenting, and accounting for a constantly changing set of interrelated tasks as do other groups of people developing complex technology. 4c1a

we constantly face more opportunities for changes or additions to our evolving system than we have resources to carry out. Therefore we must find ways to obtain as effective utilization of our ideas, and of our people, system, and material resources as we can so as to make the best progress toward our goals. 4c1a1

planning requires a framework within which information about goals, needs, possibilities, resources, and related dialog can be recorded, studied, and modified usefully. 4c1a2

ARC planning and task activity is currently conducted in the LINAC operational framework outlined below, see--,4e3). 4c1a3

The result of such coordinated analysis is the adoption of a current visible plan, or "baseline" of expected events, agreed upon system developments, their external configurations, and resource allocations. 4c1a4

The information relative to the planned system developments is contained in our Baseline Record.

The Baseline Record is a special subcollection of the Journal. It consists of a series of files specially formatted to contain task and resource allocation information, including particularly files of plans, specifications, analyses, designs, etc. 4c1b

The basic objectives of the Baseline Record System are: 4c1b1

1. To provide a central place for recording Baseline data in an organized way.

2. To prepare useful views of such data.

3. To provide a system for updating the Baseline data base.

The main responsibility for the data actually being complete and current resides with the pushers for the various tasks and activities.

some BRS design criteria are: 4c1b2

Users' opinions should be gathered and brought into the BRS system design process as it progresses.

Data input must be easy for task initiation - whether for tasks agreed upon as officially "on the Baseline of planned tasks" or just as possibilities (needs) up for consideration.

Data should be stored in a readable format to permit scanning for clerical proofing purposes, user-browsing, with flexible, but strictly formatted, storage for automatic processes to access and use in preparation of routine views and summaries of the information.

Views must be "easy" to generate - both by the operations people and by individual ARC users wanting special views.

Routinely produced views must be meaningful and useful to a wide range of users' needs.

Users must be guided - trained - in the use of the BRS, probably on a continuing basis.

The Baseline Record is composed of the portion of our currently accurate working records that represents our best definition of: what tasks we plan to perform, how we plan to do them, and how we will allocate resources (people, system service, materials). 4c1b3

This record is produced from central planning data contained in online files at ARC, and will contain various views of that information as needed to give meaningful representations of our situation.

A basic set of Baseline record views we will use includes:

(1) Schedule: by activity grouping (NIC,DSS,CSO)

(2) Schedule: all tasks by ARC planning stage

(3) Schedule: all tasks by person

(4) Baseline record summaries by task, formatted as "status" reports, with elements such as:

Information:	(about nature of task and agreements)
Buyer(s):	(for whom or what task is this task being performed)
Requirements:	(agreed upon needs this task will fulfill and certain design criteria as needed)
Design:	(details of design--or links to such--user interface features, internal implementation)
Milestones:	(significant delivery/evaluation points used when relevant)
Subtasks:	(smaller segments made visible for more detailed planning purposes as needed)
Subcontracts:	(other tasks initiated in direct support)

We have been keeping some or all of the Baseline Record information within a specially organized subcollection of the Journal, shelved separately. We will use as a "Shelf List" a topically organized Table of Contents. 4c1b4

Sections of the Baseline Record that are superseded by new Journal entries will be separately shelved with other obsolete documents.

Changes in requirements and designs will be approved and recorded as in configuration management of hardware designs.

we plan to develop new tools to aid analysis of estimates, schedules, and staff involvements, with interactive factor adjustment features to permit consideration of the effects of potential changes in configurations of dates, people, and interdependent tasks. 4c1b5

Present Baseline Record System 4c2

The present Baseline Record system has concentrated on the recording of information relevant to individual tasks being performed or under consideration by various ARC staff members. 4c2a

There now are over 200 tasks of various magnitudes to consider in our planning and operational environment at any point in time. These range from simple bug-fixing tasks to complex design or implementation tasks that may be performed by several people over many months. 4c2a1

Baseline Task Estimates as of 26 APR 72 for: JCN

Months: > Apr--May-Jun-Jul-- <				
ONR Annual Report 8622>	!xxx !	1:1	5:2	<DCE JCN JBN???
Stacking Chairs>	!xxx !	1/28	5/8	<DVN? JCN BER
Develop RADC APC Baseline>	!xxxxx !	---	5:1	<PXR JCN DVN
Projection TV>	!xxxxxx !	1:1	6:1	<EKV MLH JCN?
Demonstration Training>	!xxxxxx !	1:1	6:1	<JCN DVN MFA ???
RADC Final Report 8157>	!xxxxxxx !	1:1	6:2	<DVN JCN MFA CHI WHP RWW NGL WSD DCE
Resource Accounting Design>	!xxxxxxxxx !	---	6:1	<JCN EKV RWW DCW
Operations Development>	!xxxxxxxxxxxxx!	---	???	<JCN
NIC Operations Coordinator>	! !	???	???	<RWW JCN DCE
BRS Design>	!----------!	---	---	<PXR JCN JDH
Recruiting>	!----------!	---	---	<JCN CHI DCE RWW EKV ???
Journal Catalog>	!----------!	---	---	<JCN HAH WSD
User Documentation Maintain>	!----------!	---	---	<MFA JCN CP BER PML
Management And Coordination>	!----------!	---	---	<DCE JCN RWW WHP
Accounting>	!----------!	---	---	<JCN DVN
Visitors>	!----------!	---	---	<JCN ALL
Vacations>	!----------!	---	---	<ALL
Needs'Possibilities				
RINS NP>	! !	???	???	<JBN JCN RWW
Cat Data Element Revise>	! !	???	???	<JBN JCN RWW
RINS Entry Conventions>	! !	???	???	<JBN JCN
External Collaborator Coord>	! !	???	???	<DCE JCN

FIGURE 8. The tasks of one person as printed by the Baseline Record System.

4c2a2

we have developed a set of programs with an initial data storage system that organizes information recorded about these tasks with features that permit routine summary views

to be produced and that also make available flexible user-created views of the Baseline task information. 4c2a3

procedures have been developed for data collection and input and for view production that aid in weekly updating of the Record. These views are produced in hardcopy and are also entered into the Journal. 4c2a4

Higher-Level Planning Needs 4c3

Although we have started using ARC's Baseline Record System on a current task-by-task basis during the past year, we still need to develop a more complete, "higher level" picture of what new ARC system developments (functions, features, stages..) we want and expect to see. Among other considerations, this includes better definition of activity goals. 4c3a

Plan needs 4c3b

We are now working on a set of descriptions of proposed developmental stages for each of our activities. 4c3b1

Because our activities are strongly affected by the developments (features, timing, resource use) of others, it is clear that realistic plans for each activity will be produced only after considerable integration and adjustment.

Plans needed and who makes them 4c3c

The pusher (or a prospective pusher) for each activity is the person responsible for seeing that the developmental plan is made and kept up -- as a continuing part of his role as pusher. Thus, for example, the DSS pusher will pull together the various needs and possibilities about how the DSS should and might develop, over the coming months and years. 4c3c1

He is expected to draw upon others (including his DSS planning team) for help, ideas, or other inputs in the process, but he is the one responsible for producing the plans we need.

Rather than just getting help from others individually, he may find it may useful to have some group discussions among appropriate people for each main activity. The pusher should make this happen where needed.

Each activity plan requires many hours of effort on the part of the pusher -- particularly with the balancing and adjusting that may be needed. 4c3c2

Plan elements 4c3d

SRI-ARC 8 JUNE 1972 13041
Team Augmentation
Baseline Records System

The following eight items are basic considerations pushers will provide in their plans: 4c3d1

1. Basic objectives of the activity.

   What should it result in or produce?

2. New or changed features that may be added...including descriptions of what they are, how they might work, what they mean to the system and/or the users.

   These may be thought of either as separate tasks, or simply as "features" -- which might result from several tasks.

3. The non-machine methodology, procedures, and training that need development to really use the tools and features to produce useful total packages -- sub-systems.

4. Stages of development -- logical combinations of features, procedures, training (not just points in time, describing the "look" at significant points.

   The stages should fit the natural progression of the activity -- not necessarily related to ARC overall stages.

   Some activities will have less apparent need for showing stages of development than others. Still, it seems it is important to "partition" the future plan in some way, even if on an arbitrary, less meaningful basis.

5. Relationships to other tasks or features needed.

   Where critical needs (for each activity) exist, they will be pointed out -- with some discussion of the situation.

6. Effort needed to meet stages.

   ROUGH estimates in man-weeks by feature or stage (plus skill types or people being considered to work on it if known) are needed.

7. Alternative possibilities for other features or stages.

8. Implications on the staffing skills and levels required of ARC as a whole.

Comments on our Experience with the BRS to Date 4c4

Considering our initial experience using the initial BRS, we feel that our ARC users were not well guided and trained in BRS use. 4c4a

The initial system did not produce views that were useful enough - mainly because most of the needed data were not in the system. 4c4b

Key missing data were requirements, designs (or links to them) partly because they did not exist, partly because of a lack of participation by the user population. 4c4b1

We still need to develop better estimating techniques. The accuracy of estimates needs improvement and what estimates mean to us needs description. ARC people need to learn more about how to make predictions of start, end and other dates, resource use estimates in our changing, quite unpredictable environment. 4c4b2

A BRS-integrated accounting and resource allocation system is needed to aid in estimating, and in the decision processes in Baseline management. 4c4b3

Developing a system for the facilitation of input of data is a real challenge, but must be worked out.

An activity and task accounting number system that will be shared with the BRS has been designed. It is open-ended and will lend itself to overlapping task, activity interests.

BASIC NLS 4d

Basic NLS User Features 4d1

Introduction 4d1a

ARC focuses on the evolutionary development of the Online System (NLS) in the spirit of bootstrapping which has been applied since the project's inception. 4d1a1

Continuing evaluation based on our experiences generates the need for and the form of modifications to NLS. The tools of earlier versions of NLS are used to design and implement new versions which differ in new features and in the growth, modification, and possibly deletion of older features. 4d1a1a

We try out tools in the hope they will improve the working abilities of the group. Changes are evolutionary and small to minimize the shock to the whole system. Modifications are, however, constantly being made. 4d1a1a1

Examples of some changes to NLS and the reasons for the changes include: 4d1a1a2

the addition of the split screen display mode to make possible multi-file viewing and cross-file editing.

the removal of the trails feature because it was not used extensively.

the modification of the substitute command to provide a larger, more useful variety of parameter modes.

Our augmentation system provides a workshop of online tools and human interaction techniques used not only in software design and development, but also in the management of the group, in the operation of the Network Information Center, and will be used in the creation of online communities of discipline-oriented researchers. 4d1a1b

Our experiences in the development of augmentation system

features within the Center and on the ARPA Network indicate some new directions for our bootstrapped research effort. 4d1a2

In the contract period, emphasis has shifted from the development of tools to augment individuals toward development of tools for local project teams and also scattered communities of researchers. 4d1a2a

Such tools include: 4d1a2a1

the Dialog Support System (DSS), and

the Baseline Record System (BRS).

The first scattered community will be composed of system designers aided primarily by the Software Engineering Augmentation System (SEAS) discussed below see--,4d2). This community will collaborate in the development of a system design discipline. The augmentation of the Software Engineering community will accelerate evolution of new tools. In the future, other communities will receive specialized tools developed by the augmented system designers. 4d1a2b

In the past contract period many additions and modifications were made to NLS. A new and effective typewriter version (TNLS) has found wide use both at ARC and at sites on the ARPA Network. Improvements have been made in the display version (DNLS), and a first PDP-10 version of an offline mode (DEX) has been introduced. 4d1a3

As of February 1971, an initial version of TNLS (Teletypewriter NLS) was fully operational on the PDP-10. One of the primary reasons for its development was to fill in the spectrum of augmentation tools to be made available at less expensive hardware and computer resource costs than are necessary to run a DNLS system. 4d1a3a

There are currently many people over the ARPA Network who use the system in their work. The TNLS command set is largely synonymous with DNLS, barring features peculiar to the display (e.g., Split Screen) and most of the recent features available in DNLS are available in TNLS (e.g., Sort Merge). 4d1a3b

The basic differences between the command vocabularies

of TNLS and DNLS are in the area of addressing. DNLS is a highly interactive, nonlinear, visual system while TNLS, owing to the nature of the medium, is less interactive and linear. In an effort to compensate for the deficiencies of the medium, many special TNLS addressing features have been made available to the user. 4d1a3c

It should be noted that the TNLS command and addressing language is richer than that of most other "text editors"; some would accuse it of being confusing. Novices, however, can quite effectively start by using a subset of the features. 4d1a3c1

The system, as with all systems developed at ARC, is meant to provide a workshop of tools to many levels of user experience to aid in the augmentation of intellectual tasks. Thus, making use of various combinations of address specifications, the sophisticated TNLS user may accomplish the equivalent of crossfile editing. 4d1a3c2

A new TNLS guide has been written (see -- 7470,), reproduced, and distributed to Network and local users. This guide contains a complete description of TNLS commands and Journal, Identification, and Number System commands in both detailed and summary form. It is designed so that as the system evolves, it can be easily updated so as to remain current and useful. 4d1a3d

Several training courses for Network users of the NIC and TNLS have been held. They are described in this report as part of NIC activities (see--,5g10a)). 4d1a3e

New special purpose subsystems (in addition to the Dialog Support System (DSS) and the Baseline Record System (BRS) described elsewhere in this report) have been developed or improved. These include a sort-merge system and a user program system among others. 4d1a4

NLS -- Technical overview 4d1b

Introduction 4d1b1

The current implementation of NLS on the PDP-10 is a

large, continually evolving program. Code presently occupies about 150,000 words of computer storage. 4d1b1a

This section presents an overview of the organization of NLS and the structure of files in the system. 4d1b1b

Descriptions of earlier versions of NLS may be found in previous ARC reports. The April 1970 report (5139,) contains a detailed discussion of the system as it existed in its final days on the XDS-940. 4d1b1b1

Changes have been made in the logical structure of the system for several reasons: 4d1b1b2

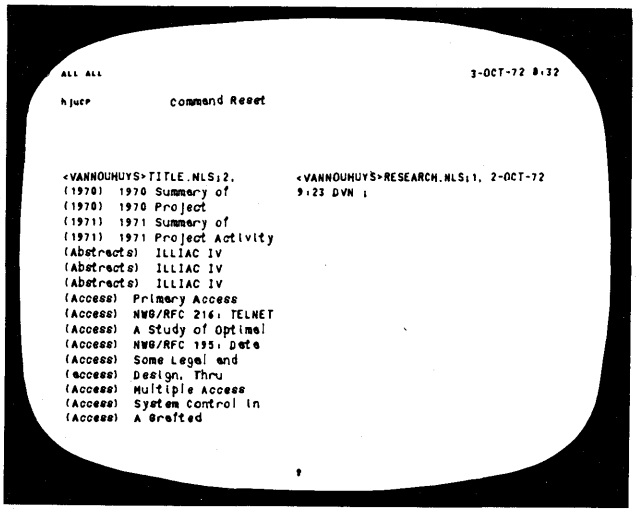


FIGURE 9. Cross file information manipulation with split screens: initially the screen is split with a title index on the left and a new empty file named research on the right. Titles are truncated to show more lines.

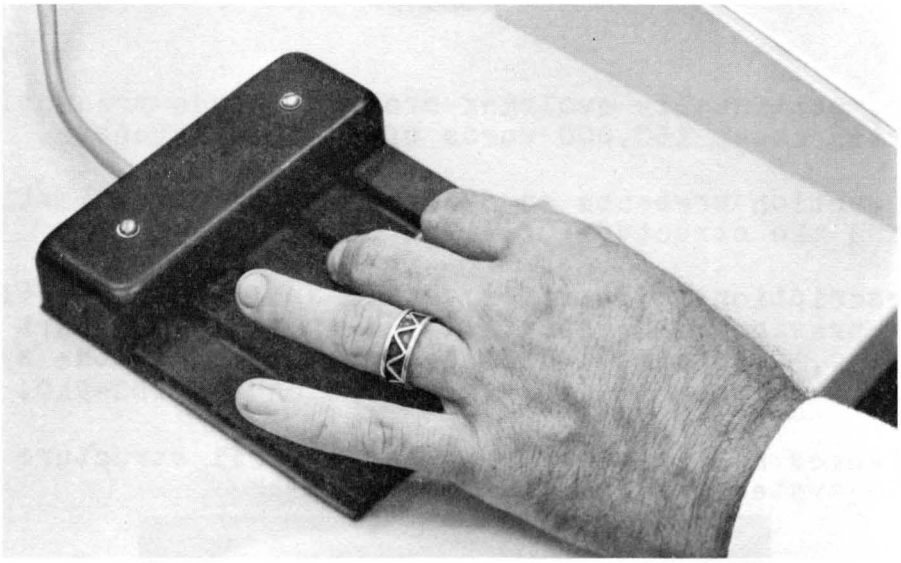


FIGURE 10. User calls the Goto Programs Subsystem by entering the first letter of each command via a keyset. On the keyset he can enter with one hand all characters that can be entered from a keyboard. The other hand is free to use the mouse (below).

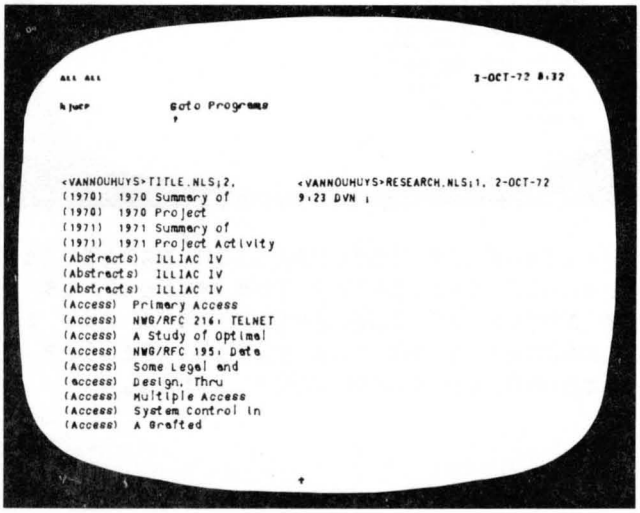


FIGURE 11. The Goto Programs Subsystem (see command feedback line in the upper middle of the photo) includes several user service subsystems.

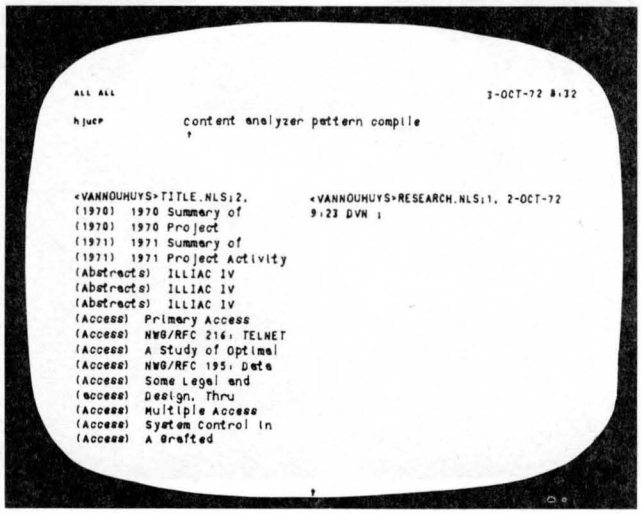


FIGURE 12. He calls one of the user systems, the content analyser.

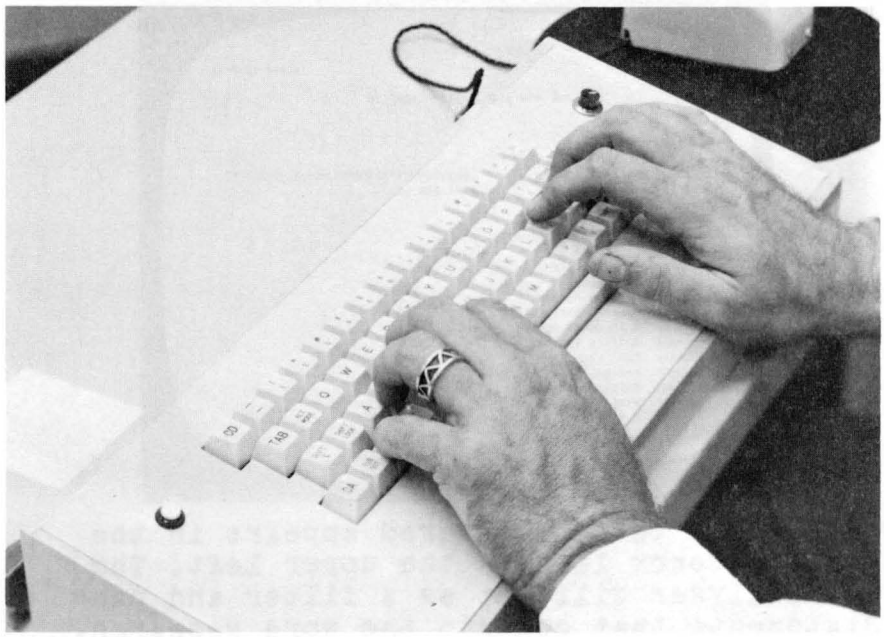


FIGURE 13. To insert text of more than a few characters, this user switches hands to the keyboard. Practice varies among users.

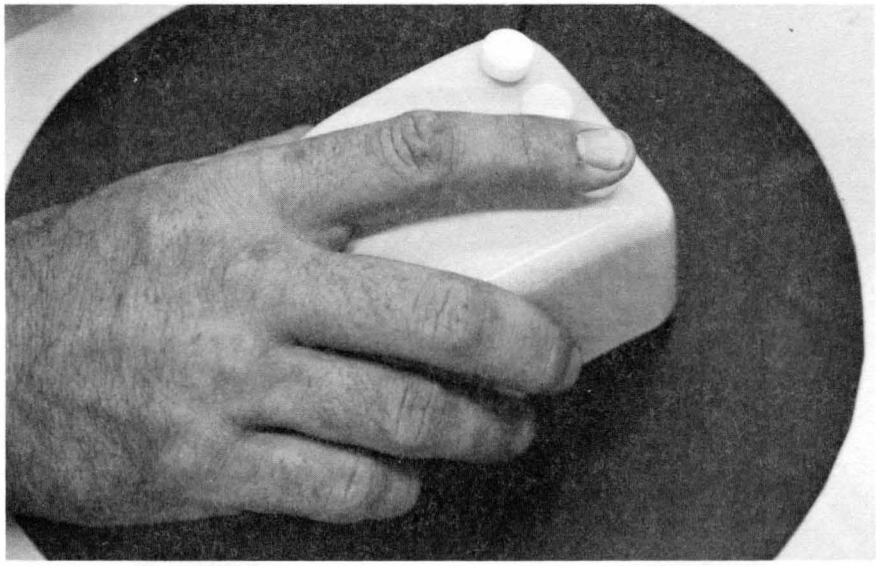


FIGURE 14. The user returns his left hand to the keyset and his right hand to the mouse. To confirm his command to compile the pattern "Research" he presses the right hand button on top of the mouse.

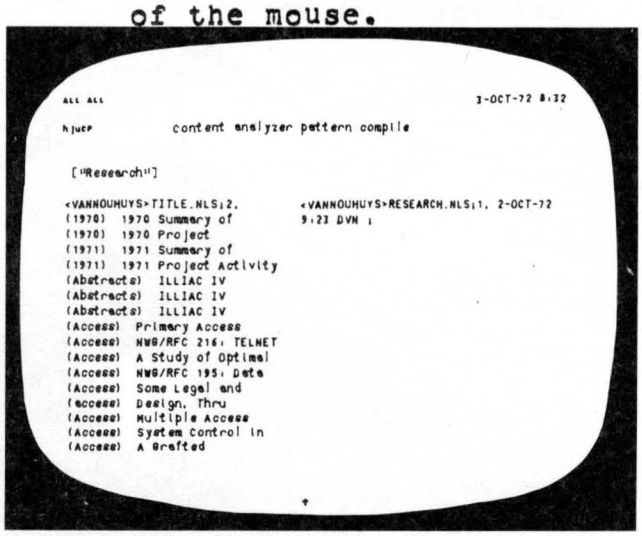


FIGURE 15. The text he entered appears in the literal feedback line on the upper left. The content analyser will act as a filter and pass only statements that contain the word Research. Patterns may be much more complex and include logical operators and classes of characters as well as specific characters.

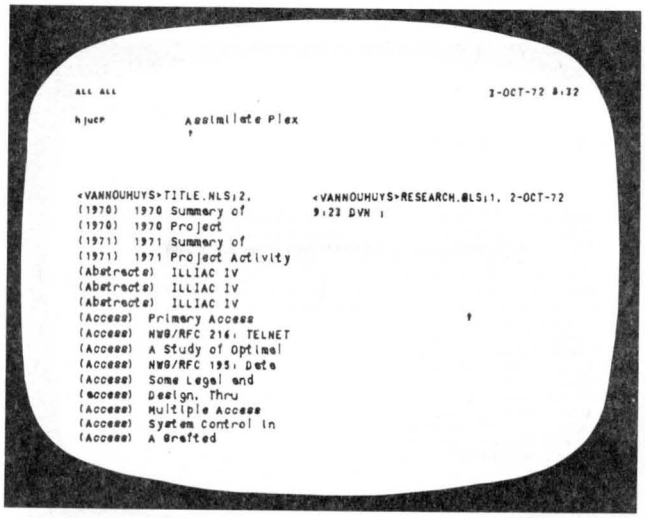


FIGURE 16. The user calls the command Assimilate Plex by entering three characters on the keyset. At the same time he is rolling the mouse which moves the arrow to the area (middle right) where he wants to put the assimilated material.

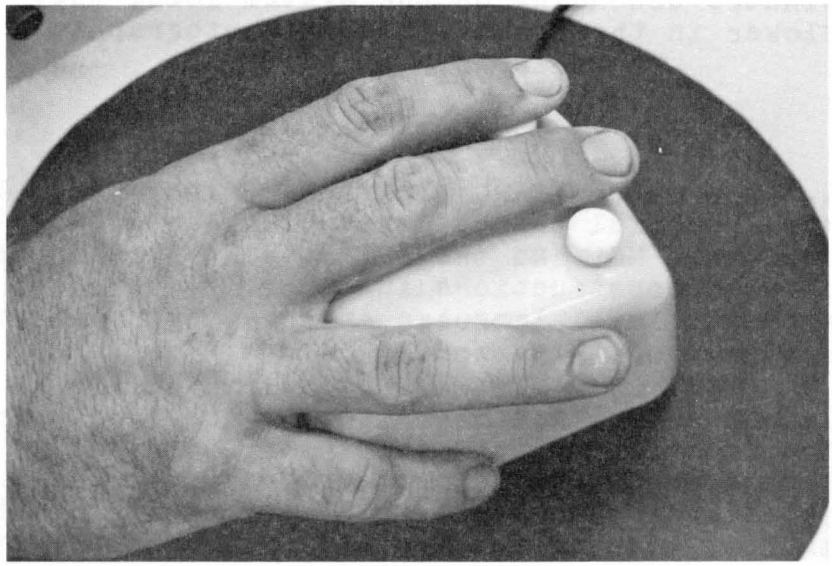


FIGURE 17. By holding down the left hand two buttons on top of the mouse, he commands the system to assimilate only items that pass the content pattern he specified above. The letter i toward the upper right and the larger letters on the upper left are feedback from this command.

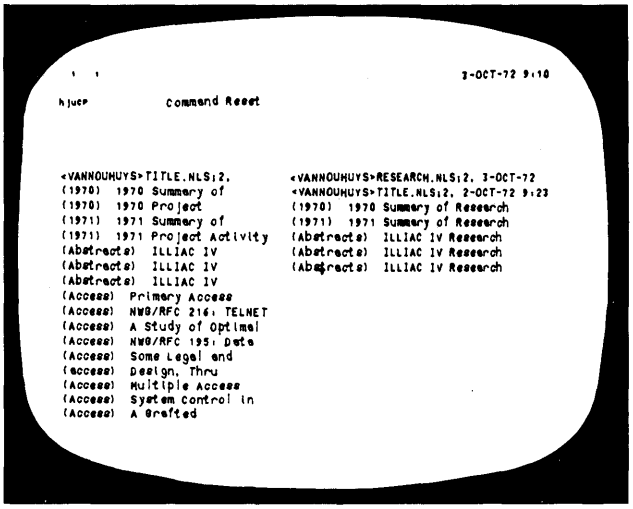


FIGURE 18. Assimilate has created a new file of papers on Research. Such steps normally take from 1 to 5 minutes depending on the system load; they were slower in these examples for photographic reasons.

1. The current ARC programming language, L10, is more powerful than the several languages it replaces, MOL and the SPL's. L10 permits special purpose constructions anywhere in its code. It is a higher level language and provides greater compiler optimization.

2. An effort has been made to modularize further the functions within the system to ease development by a team of programmers. This functional modularity will be increased with the introduction of the Modular Programming System, see--,4d2f).

Discussions of the user features of the systems and subsystems making up NLS may be found in the following locations: 4d1b1c

DNLS: See DNLS user guide. (10703,) 4d1b1c1

TNLS: See TNLS user guide. (7470,) 4d1b1c2

DEX: See DEX user guide (9934,) and below (see --,4d1d). 4d1b1c3

Journal: NIC Journal User Guide (7635,) and see --,4a) and --,5f2a) of this report. 4d1b1c4

Identification: See TNLS user guide (7470,) and see --,4a4) in this report. 4d1b1c5

Catalog: See --,4a6b2) and --,5g3). 4d1b1c6

Sorter-Merger: See ---,4d1e1). 4d1b1c7

Baseline: See --,4c). 4d1b1c8

NLS-DDT: See --,4d1e4a6). 4d1b1c9

NLS file structure 4d1b2

Introduction 4d1b2a

The format and structure of NLS files were determined by certain design considerations. 4d1b2a1

It is desirable to have virtually no limit on the size of a file. This means it is not practical to have an entire file in core when viewing or editing it.

The time required for most operations on a file should be independent of the file length. That is, small operations on a large file should take roughly the same time as the same operations on a small file. The user and the system should not be penalized for large files.

In executing a single editing function there may be a large number of structural operations.

A random file structure staisfies these considerations. Each file is divided into logical blocks that may be accessed in random order. There are several types of blocks, each with its own structure. 4d1b2a2

An NLS file is made up of a header and up to a fixed number (currently 465) of 512-word file blocks. 4d1b2a3

File Header 4d1b2b

File header contents: 4d1b2b1

- File creation date
- Version word (changed when NLS file structure changes)
- Identification of last user to update or output the file.
- File owner.
- Left name delimiter default.
- Right name delimiter default.
- Number of structure pages used.
- Number of data pages used.
- Status table -- One word per ring block or data block page. Contains the following:
  - Whether page has been modified by a user.
  - Free space count (for data block)
  - Pre-garbage collection count.
  - Post-garbage collection count.
  - Free list pointer (for ring block)
- Marker table.

Structure Blocks -- ring elements 4d1b2c

These blocks contain fixed size ring elements with a free list connecting those not in use. 4d1b2c1

Ring element contents: 4d1b2c2

- Pointer to first substatement.
- Pointer to successor statement.
- Pointer to the SDB that contains text for this statement.
- DEX work area.
- Head of plex flag.
- Tail of plex flag.
- Name flag.
- Name hash.
- Statement identifier and free list link.

Data Block -- statement data blocks 4d1b2d

Data blocks are composed of variable sized blocks called Statement Data Blocks (SDB's) which contain the text of NLS statements. New SDB's are allocated in the free space at the end of a data block. SDB's no longer in use (because of editing changes) are marked for garbage collection when the free space is exhausted. 4d1b2d1

Statement Data Block (SDB) header contents: 4d1b2d2

- No-longer-used SDB flag.
- Length of SDB.
- Length of string in SDB.
- Left name delimiter.
- Right name delimiter.
- Pointer to ring element.
- Length of name.
- Last write time.
- Last write ident.

String Identifiers and Text Pointers 4d1b2e

A string identifier (STID) is a data structure used within NLS to identify strings (possibly within NLS statements). 4d1b2e1

If the string is in an NLS statement, the STID contains a file identifier and a ring element identifier.

The presence of a file identifier within the STID all editing functions to be carried out between files.

Text pointers are used with the string analysis and construction features of L10. They consist of an STID and a character count. 4d1b2e2

Locking mechanism -- Partial copies 4d1b2f

The NLS file system under TENEX provides a locking mechanism, which protects against inadvertant overwrite when several people are working on the same file. Once a user starts modifying a file, it is "locked" by him against changes by other users until he deems his changes consistent and complete and

issues one of the commands: Update File, Output File, or Unlock File which "unlock" the file. Note, a user can leave a file locked indefinitely -- this protection is not limited to one console session. 4d1b2f1

When a file is locked (is being modified), the user who has modification rights sees all of the changes that he is making. However, others who read the file will see it in its original, unaltered state. If they try to modify it, they will be told that it is locked by a particular user. Thus the users can negotiate for modification rights to the file.

This feature is implemented through the use of flags in the status table in the File Header and through the partial copy mechanism. 4d1b2f2

All modifications to a file are contained in a partial copy file. These include modified ring elements and SDB's.

Core Management of File Space 4d1b2g

When space for more data is needed, the following steps are taken in order until enough is found to satisfy the request: 4d1b2g1

1. Core-resident pages are checked for sufficient free space.

2. Other pages are checked for free space. If one has sufficient space, it is brought in.

3. If garbage collection on any page in the file will yield a page with sufficient free space, then the page which will give the most free space is brought into core and garbage collected.

4. Otherwise a new page is created.

Logical structure 4d1b3

Introduction 4d1b3a

Interaction support 4d1b3b

Terminal interaction support 4d1b3b1

Display interaction support

The display interaction support routines take input from display users, support various L10 display input constructions which allow the creation of simple interaction statements, and control the command feedback line, name area, view spec area, and bug selection areas of the display screen.

Typewriter interaction support

The typewriter interaction support routines are primitives for interacting with a typewriter terminal user. They include input, command feedback, literal collection, and error feedback routines.

Sequential file input support 4d1b3b2

Sequential file input support routines take input from DEX sequential files or a control file and pass it to the DEX subsystem processor or the control file driver system, respectively.

Subsystem control 4d1b3c

Command specification 4d1b3c1

The command specification routines receive information from the input interaction level or sequential file input and process it as follows:

1. Command mnemonic input from the user is parsed using tests implemented as a large set of nested case statements which check successive command characters.

2. Operands for commands are interpreted where necessary.

3. Control is transferred to the appropriate execution routine.

4. Control is transferred to the Portrayal Generator for formatting and display.

5. The user may repeatedly execute commands of a given type with different parameters by specifying more parameters. When the user types a character which can not be a parameter specification, the input is assumed to be a new command.

At any time prior to execution, the user may abort an individual parameter specification and enter a corrected operand without destroying operands previously entered in multi-parameter commands. It is, however, possible to abort an entire command at any time before it is executed.

Subsystem Support 4d1b3c2

These routines support the parsing of particular subsystems and provide the code necessary to translate the high level functions of each subsystem into calls on the file manipulation and portrayal generation routines of NLS. They also have code necessary to implement any additional facilities needed by the subsystem.

Portrayal Generator 4d1b3d

Display control 4d1b3d1

The display controller is composed of

1) a fast formatter and data structures that allow NLS to modify portions of the display image in response to user modification of the files being displayed, and

2) user controls, such as the DNLS jump

commands, over what is portrayed and how much is shown.

This formatter can maintain images in several "display areas" at one time, updating them as necessary. Each area may display information from several files.

Typewriter terminal print control 4d1b3d2

This is a formatter that is oriented toward printing parts of a file onto a typewriter terminal.

Hardcopy formatters 4d1b3d3

These include a relatively simple system, Quickprint, and a more complicated formatting program, the Output Processor.

Quickprint formats the text for printing as it appears through the display or typewriter terminal formatters.

The Output Processor can feed to a variety of different devices, including printers and microfilm, and controls the formatting of the document according to directives embedded within the text. For details, refer to the "Output Processor User Guide",(,11076,2).

Sequence generator 4d1b3d4

Succeeding calls on the sequence generator create a sequence of statements which satisfy system or user filters starting at a place in the file specified by the user.

An example of the system filters it observes in deciding whether the identifier of a statement should be part of a sequence is the level truncation viewspec which permits the display of only those statements above particular levels in the NLS hierarchical file structure.

These sequences of statement identifiers are used

by formatters for terminal or hard-copy portrayal, by compilers, or by processors which manipulate files, such as the sorter.

See--,4d1e4) for a discussion of the sequence generator with user programs.

User filters and reformatters 4d1b3d5

The user may write and incorporate additional filters which the sequence generator will use as a final acceptance test. These user-supplied filters may reformat the text of the file for special applications or views.

User sequence generators 4d1b3d6

The user can write his own sequence generators which can make use of any NLS routines.

Editing 4d1b3e

File manipulation algorithms 4d1b3e1

These algorithms carry out the file manipulation commands of NLS. They decide what is to be done by the textual and structural editing routines and in what order. Utility routines actually manipulate the NLS files.

Some commands make use of textual editing routines exclusively (e.g., "Insert Text"); some use only structural editing routines (e.g.,"Move statement"); others use a combination of the two (e.g., "Insert statement").

These algorithms can move and copy text from one file to another through cross-file editing..

Structure editing 4d1b3e2

These routines involve the manipulation of ring structure alone and do not alter the contents of the statement data blocks which contain the text.

Text editing 4d1b3e3

These routines edit the text of NLS statements. Content analysis features of L10 are used to determine where changes should take place; the string manipulation and SDB manipulation machinery then change the contents of the file.

Special purpose processors 4d1b3f

Inserting and outputting sequential files 4d1b3f1

These processors create NLS files from sequential files and vice versa.

Compilers 4d1b3f2

Currently four compilers are available from NLS. In addition we are now studying ways of making available through NLS the assemblers of the TENEX operating system.

The four compilers now available are:

L10, a procedure-oriented, block structured language developed by ARC for use on the PDP-10,

A subset is available as the content analyzer. (9246,10) and see --,4d1e4d) in this report.

IMOL, a procedure-oriented, block structured language which produces code for the IMLAC computer-display.

Tree-Meta, a compiler-compiler used by ARC staff to develop other languages, such as L10 and IMOL. (See the Tree-Meta Report (10869,,) and --,4d2e) of this report.)

MPL, the Modular Programming Language, an experimental new language to be used to rewrite NLS. (See --,4d2f)

Text is passed to these compilers through the

sequence generator (and thus can be filtered and reformatted enroute to these various processors).

Utility routines 4d1b3g

NLS file system 4d1b3g1

These routines implement and manipulate the data structures in Tenex files which NLS uses. Unlike other routines discussed above, they are cognizant of and deal with the data structures and the TENEX timesharing system environment.

They are responsible for:

Opening and closing files.

Managing the portion of core set aside for file pages.

Writing on and reading from files.

Manipulating ring elements and SDB's.

Moving within the NLS file structure by following ring element pointers.

Statement name lookup.

NLS string system 4d1b3g2

Supports string manipulation construcions in the L10 language and deals with the NLS Statement Data Block and Ring Block structure.

Miscellaneous support routines 4d1b3g3

Basic L10 language support routines.

Call mechanisms.

Display support routines 4d1b3g4

Information writing on the screen.

Manipulating information on the screen.

NLS -- New features 4d1c

The following features, common to both DNLS and TNLS, are new on the PDP-10: 4d1c1

Name Delimiters 4d1c2

A user may specify the characters to be used for left and right name delimiters for statements within any structural entity in an NLS file. The system defaults are left and right parentheses. 4d1c2a

Jump to Word/Content 4d1c3

The user was provided with the capability of jumping to the first or next occurrence of a specified word or text string. 4d1c3a

Null File 4d1c4

A new command, Null File, has been added to TNLS and DNLS. Given a file name, it will create an empty NLS file with that name. Upon completion of the command the user is left with the CM (Control Marker -- TNLS) / display start (DNLS) at the origin of this new file. 4d1c4a

Output Assembler 4d1c5

Sequential files acceptable to the DEC assembler may be created from NLS files using this command. 4d1c5a

Output Compiler 4d1c6

The capability to drive TREE-META produced compilers (including the L10 language compiler) directly from NLS files is available. 4d1c6a

Output Sequential 4d1c7

The user may produce a sequential file that corresponds to his NLS file. Spaces are used to indicate the level of a statement. 4d1c7a

Insert Sequential 4d1c8

The Insert Sequential File command converts sequential files into NLS format. This also allows the user to convert XDS-940 files to TENEX-NLS format. 4d1c8a

Output Quickprint 4d1c9

Since users often want quick hard copy of their files, the Output Quickprint command was added. Unlike the Output Processor, this formatter does not make use of embedded formatting directives. The command offers the user a default file name and a default of 1 for the number of copies to print; these may be superseded by the user. After the document is formatted it will be automatically spooled for printing. Viewspecs in effect at the time the command is given control the format and content of the printed text. 4d1c9a

Update File -- File Locking 4d1c10

The NLS file system under TENEX provides a locking mechanism, which protects against inadvertant overwrite when several people are working on the same file. Once a user starts modifying a file, it is "locked" by him against changes by other users until he deems his changes consistent and complete and issues one of the commands: Update File, Output File, or Unlock File which "unlock" the file. Note, a user can leave a file locked indefinitely -- this protection is not limited to one console session. 4d1c10a

When a file is locked (is being modified), the user who has modification rights sees all of the changes that he is making. However, others who read the file will see it in its original, unaltered state. If they try to modify it, they will be told that it is locked by a particular user. Thus the users can negotiate for modification rights to the file. 4d1c10a1

The users are also allowed to enter "Browse Mode", which allows several users to simultaneously modify a file. When they leave browse mode, one of them may elect to keep his changes if no one has the file locked, in which case he locks the file until an update or output command is executed by him. 4d1c10b

Goto Exec 4d1c11

The user may start a new copy of the TENEX EXECUTIVE below NLS in the job's process structure and execute arbitrary EXEC level commands, including running other subsystems. Then, by issuing the EXEC quit command, the user is returned to NLS, exactly as he was before issuing the Goto Exec command. 4d1c11a

Execute Logout 4d1c12

The new Execute Logout command is equivalent to issuing the Execute Quit command in NLS and following it with a LOGOUT command in the EXEC. 4d1c12a

The following features in DNLS are new on the PDP-10: 4d1c13

Split Screen and Cross File Editing 4d1c14

Display Screen Splitting and Formatting 4d1c14a

FIGURE 19. (Opposite) Overall NLS logical structure.

4d1c14a1

Goto Display Area Control 4d1c14a2

Horizontal Split

This splits the display area in which the BUG occurred horizontally (into an upper and lower segment) at the bugged location moving the image of the original display area to the upper or lower segment depending on whether the cursor is above or below the bugged position when the final CA is input.

No display area will be created which is smaller than 2 lines by 20 columns (using the character size of the original display area).

Vertical Split

This splits the display area in which the BUG occurred vertically (into a left and right segment) at the bugged location moving the

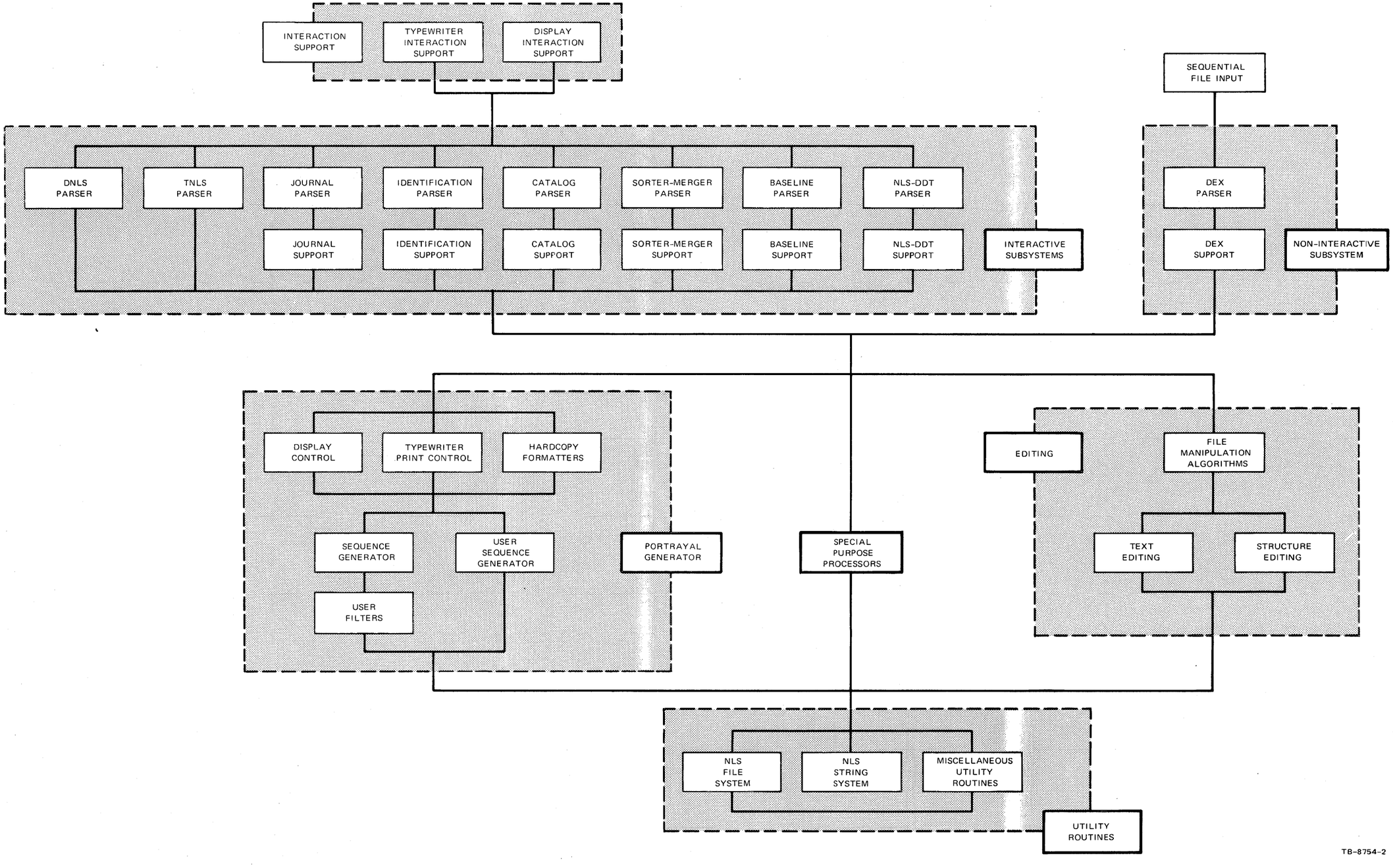


FIGURE 19 OVERALL NLS ORGANIZATION

image of the original display area to the left or right segment depending on whether the cursor is to the left or right of the bugged position when the final CA is input.

Move Boundary

The selected boundary (first BUG) is moved to the new position (second BUG). A boundary will not be moved past a boundary of a neighbor. A boundary is moved for all display areas for which it is a boundary. Any resulting display area which is smaller than 2 lines by 20 columns will be deleted.

Format Display Area

Character Size

The current character size of the display area that currently contains the cursor is displayed, and the user may type a number (0, 1, 2, 3) for a new character size. Different display areas may simultaneously have different character sizes.

Clear Display Area

The bugged display area is cleared, i.e., the image is erased, the return and file return rings are released, and the association of a file with that display area is removed. The display area itself is not deleted.

Cross File Editing in DNLS 4d1c14b

One may freely edit and jump using several display areas. The position of the cursor is used to resolve ambiguities. 4d1c14b1

For example, if one executes a Jump command, the position of the cursor when the final command accept is entered determines in which display area the new image is to appear.

Also, if one changes viewspecs using the leftmost

two buttons of the mouse, the viewspecs of the display area containing the cursor when the buttons go down are used as the initial values and are displayed in the viewspec area. When the buttons are released, the display area containing the cursor receives the new viewspecs.

Substitute Commanrc Change 4d1c15

Substitute in DNLS (and soon in TNLS) has been expanded to allow words, visibles, etc., to be substituted in a structural entity. 4d1c15a

All of the old basic NLS substitute commands are still available and work as before. In addition, the commands, Substitute *[text entity]* in *[structure entity]* are now available. Text entity may be Character, Word, Visible, etc., and Structure entity may be Statement, Branch, Group, or Plex. 4d1c15b

During the substitution, the delimiters of the candidates for substitution are observed. For example, if the user issues Substitute Word... "the" for "an" in the statement "Do you want an igloo instead of another kayak, dear?", the word "an" will be replaced by "the", but the word "another" will not be changed. 4d1c15c

Display Creation Efficiency Improvement 4d1c16

The former code that generated and maintained the display image updated the whole screen except in the case of textual edits. We were able to optimize this process so that, in most cases, only those windows involved in the operation are changed, and only those entities involved are actually reformatted. The response time for a display user has been reduced substantially. 4d1c16a

The Deferred Execution System (DEX) 4d1d

Deferred Execution (DEX) is a system that provides a means by which information may be prepared offline for later processing by the computer. 4d1d1

The currently running system, DEX-1, has commands that provide for text input, backspacing over characters,

deletion (and undeletion) of commands, and the creation of NLS files and hardcopy printouts. DEX-2 will provide further editing capabilities as well as access to existing files. 4d1d2

DEX-1 was designed to be used with typewriter terminals connected to some recording device (currently paper tape or magnetic tape cassette). At such a terminal the user produces a paper tape or tape cassette containing information destined for computer processing. 4d1d3

DEX is a complement to the online NLS. It operates with greater system-use efficiency since actual computer time can be deferred to periods of low usage -- "off-hours", when the load is greatly reduced. 4d1d4

The end result of files created by DEX and files created by NLS is the same. Once created by either system, no distinction is made -- they are all NLS files that may later be edited online. 4d1d5

The overall goal of DEX is to increase the utility of our computer aids by, in most cases, reducing the support cost of computer-aided text manipulation, and in some cases providing more service value to the user than he would obtain from immediate-execution processes. 4d1d6

There should be a smooth spectrum of features applicable to different situations of service level, terminal device, information context and type or priority of task. 4d1d6a

In such a spectrum of computer aids, users should find complete consistency and continuity in concepts, nomenclature, and operating skills required for operating effectively in these different situations. 4d1d6b

Users should eventually be able to switch from one level of interaction to another while at an online terminal thereby providing maximum utility toward the user's working goals. 4d1d6c

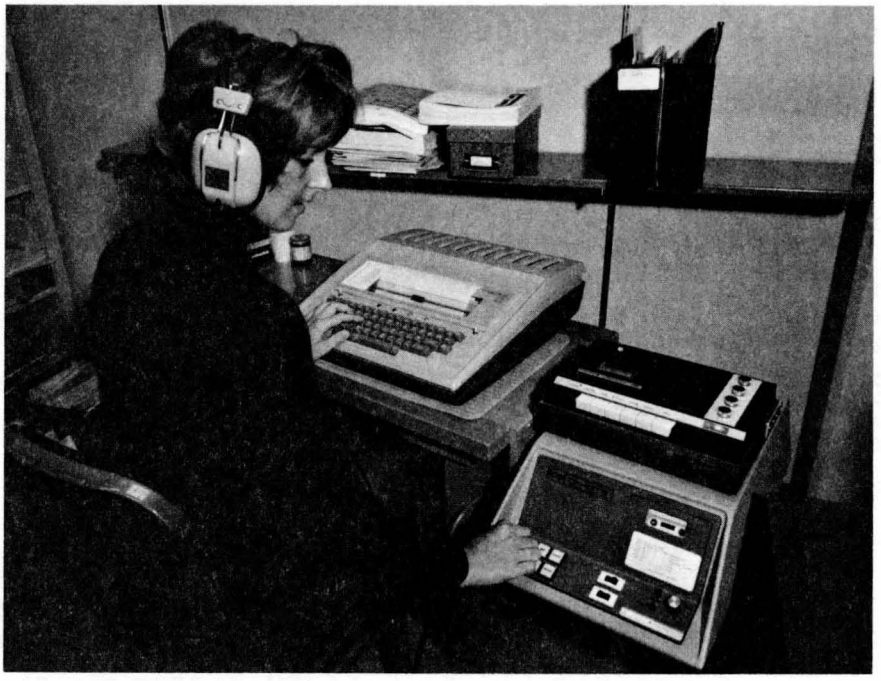


FIGURE 20. Deferred execution (DEX) operator transcribing a meeting. The black box to the operator's right plays an audio tape of the meeting. She types the words on her terminal and the terminal output is recorded by the digital tape machine that she is touching with her right hand. The digital tapes are read onto our disk storage and processed into NLS format at leisure.

4d1d6c1

DEX-1 was a first attempt at satisfying these goals and was implemented primarily to provide an offline input facility. DEX-2 will provide editing facilities and more flexible input. Later stages will make deferred features available in the online modes. 4d1d7

The design of DEX-1 was carried out in an augmented mode making use of the dialoging possibilities of the Journal. DEX-2 has been designed using these same capabilities with a team approach. Thus a record of the system from first ideas to final documentation is available. The implementation of DEX-2 is expected to proceed soon. 4d1d8

A manual for DEX-1 is currently available. (9934,). 4d1d8a

The design for DEX-2 is documented in (9241,). 4d1d8b

Other subsystems 4d1e

Sorter-Merger-Updater Description 4d1e1

General Implementation Description 4d1e1a

The new sort-merge-update capability is based on the addition of three primitives to NLS that are used by the Sort Branch/Plex/Group and Merge Branch/Plex/Group commands and which may also be called from user L10 programs compiled with the "Goto Programs L10 User Program Compile" command. 4d1e1a1

Each of the three primitives added to NLS to perform the sorting, merging, and updating functions requires as an argument the address of a key procedure program written in the L10 language to furnish sorting criteria. In addition, the update primitive requires as an argument the address of an update decision procedure to take action on corresponding data items. This procedure will differ for various specific applications. In the most general case, it is provided by the user, although we are building up a library of some standard procedures for common applications.

The sort primitive uses a tree sort as its basic algorithm. This is the same one used in our previous sorting system. The restriction of its application to intra-file use, the implementation of efficient key comparison algorithms and a special reordering routine have resulted in a speed increase on the order of 100.

Procedures supplied to the Sort, Merge, and Update primitives: 4d1e1o

Key procedure: 4d1e1b1

Sort key procedures are written in the L10 language and provide the patterns for text string analysis through which a data base in an NLS file is to be sorted.

Typical keys may be written to:

find and order last names after initials

find numbers in columns

find individual key words in indices

The system default alphabetizes statements over which the system is run.

Update Decision Procedure 4dle1b2

The update decision procedure is called by the update primitive once for each sort key value found in either the master or update input.

All of the statement identifiers (stid's) supplied to this procedure on a given call have the same key value as determined by the key procedure.

In general, this procedure changes the master file by deleting some branches from the master input and inserting some of the update input.

In the simplest case, there would be at most one master and/or update item for a given key value. In this case, the update decision procedure deletes the master item when there is a corresponding update item to replace it. Other master items are kept and other update items are inserted after the destination stid.

A comparison file may be created by this procedure for proof reading.

Control File and Record Mode 4dle2

A set of commands (and modifications to the user input routines) has been added to implement a record and playback capability. A session or series of operations at a display console may be recorded on a file, then played back. During the playback, NLS will read the input from the control file instead of from the user.

An attempt is made to replay the commands at the same speed that the user entered them. 4d1e2a

This allows us to capture user interaction with NLS for analysis and for creating a "control load" to use in testing the effects of changes to the TENEX and/or NLS systems. In addition, users can build up a library of common sequences of commands, which can then be executed quite easily. Also, comprehensive testing of new releases of NLS can be accomplished using such recorded user interaction. 4d1e2a1

Output Processor Addition 4d1e3

The Output Processor is an NLS file formatter, driven by embedded directives, for various output media such as a line printer or microfilm. This subsystem was expanded to provide a larger variety of directives (summarized in the "Output Processor Brief User Guide" (6912,)) and to permit such the use of the FR-80 microfilm device. 4d1e3a

The output processor subsystem code was rewritten in Tree-Meta to provide an interpreter for the formatting directive language. 4d1e3a1

FR-80 Output Processor Device 4d1e3b

Documents may again be formatted for FR-80 microfilm devices. The document formatter (commonly called the Output Processor) provides the following options with respect to this device: 4d1e3b1

64 character sizes,

placement of text within a 16k by 16k coordinate system,

various intensities and line widths, and

microfilm/fiche and/or paper output.

User Programs 4d1e4

Introduction 4d1e4a

User-written programs enable one to tailor the

presentation of the information in a file to his particular needs. Experienced users may write and compile online programs that edit files automatically. These programs, written in the L10 programming language used by NLS system programmers, may be composed using the NLS text editor, compiled into the user program buffer, and linked into the user's running NLS system. 4d1e4a1

The language contains some high level features for operations such as string analysis and manipulation which are implemented in the language as calls on NLS library routines.

The User Program facility brings together the tools formerly described as Higher Level Processes (HLPs) in the June 1971 Report (8277,). The current system provides the user with access to the full array of NLS system tools as well as the debugging facility, DDT. The ability to create what are known as User Sequence Generator programs allows greater file reordering than did the old Analyzer Formatter. User Programs also satisfy some objections to the earlier Executable Text, which could not be easily programmed or debugged.

NLS provides a variety of commands for file manipulation and viewing. All of the editing commands, and the print command with associated viewspecs (like line truncation and statement numbers) provide examples of these manipulation and viewing facilities. 4d1e4a2

But occasionally one may need more sophisticated view controls than those available with the viewspec and viewchange features in NLS. 4d1e4a3

For example, one may want to see only those statements containing a particular word or phrase.

Or one might want to see one line of text that compacts the information found in several longer statements.

One might also wish to perform a series of routine

editing operations without specifying each of the NLS commands over and over again. 4dle4a4

The Network Information Center at ARC uses the ability to create text using the information from several different statements (and even different files) and the ability to insert this new text into a file to produce catalogs and indices.

These programs may range from simple content analysis pattern filters which alter the way a file is viewed by a user to advanced programs that provide sequence generators and sort keys to edit and restructure many files automatically upon execution. 4dle4a5

Users taking advantage of this expanded feature also have access to the debugging facility of the system. Currently this means that the TENEX DDT may be used with compiled and instituted user programs (i.e., those which have been linked into the user's running NLS system). A planned expansion will make available a debugger in the NLS system itself providing an extremely powerful programming tool. 4dle4a6

While the user program tool itself has been available in various forms for several months, the complexity of the language and of the NLS internal structure have precluded any major attempt to make it generally available in its most powerful forms. Content analysis patterns have been as far as most users have gone in their use. 4dle4a7

Some non-programming personnel at ARC, however, have been creating programs to produce formatted catalogs; programmers have used the feature to create and debug new NLS commands and subsystems without being forced to compile and load the entire NLS system whenever a change is made, an inefficient and time consuming process given the demands on system resources and the current size of the system. 4dle4a8

To make this powerful tool more generally usable, an initial documentation of a subset of the L10 language has been created. This "L10 Primer" provides basic information on the syntax and semantics of many of the constructions of the whole language. It also

describes the basic commands in NLS that provide the user interface between NLS and user programs. (9246,). 4dle4a9

Omitted from the documentation are discussions of some special purpose language constructions used in the creation of NLS display commands. Also currently undocumented are system procedures that may be accessed through user programs and which facilitate building the more complex file editing and manipulation tools. Supplements to the "Primer" and the continuing documentation of the NLS system in general will deal with these omissions.

Creation of User Written Programs 4dle4b

User written programs must be coded in L10. They may call other user written routines and various procedures in the NLS program itself. 4dle4b1

User programs that control the way material is portrayed take effect when NLS presents a sequence of statements in response to a command like Print Group (in TNLS) or Jump to Item (in DNLS). 4dle4b2

In processing a command such as Print, NLS looks at a sequence of statements, examining each statement to see if it falls within the range specified in the Print command and if it satisfies the viewspecs. At this point NLS may also pass the statement to a user written program to see if it satisfies the requirements specified in that program. If the user program returns a value of true, the (passed) statement is printed and the next statement in the sequence is tested; if false, the next statement in the sequence is tested.

Although a user program may be called explicitly, user programs that modify files usually gain control at the same point in processing as those that control the view. 4dle4b3

Typically, one wants such a program to operate on a sequence of statements chosen by a user when he

decides to run the program. In addition, one usually wants to see the results of such an automated series of editing operations immediately after it happens.

Context of User Written Programs -- The Portrayal Generator 4dle4c

Generally, the user written program runs in the framework of the portrayal generator. It may be invoked in several ways, described below, whenever one asks to view a portion of the file, e.g., with a Print command in TNLS, with any of the Output to Printer commands, and with the Jump command in DNLS. 4dle4c1

All of the portrayal generators in NLS have at least two sections -- the sequence generator and the formatter; if the user invokes a program of his own, the portrayal generator will have at least one, and possibly two, additional parts -- a user filter program and a user sequence generator. 4dle4c2

Sequence Generator 4dle4c3

The sequence generator looks at statements one at a time, beginning at the point specified by the user. It observes viewspecs like level truncation in determining which statements to pass on to the formatter.

For example, the viewspecs may indicate that only the first line of statements in the two highest levels are to be output. The default NLS sequence generator will return pointers only to those statements passing the structural filters; the formatter will further truncate the text to only the first line.

One of the viewspecs that the sequence generator pays particular attention to is "i" -- the viewspec that indicates whether a user filter is to be applied to the statement. If this viewspec is on, the sequence generator passes control to a user filter program, which looks at the statement and decides whether it should be included in the sequence. If the statement passes the filter

(i.e. the user program returns a value of true), the sequence generator sends the statement to the formatter; otherwise, it processes the next statement in the sequence and sends it to the user filter program for verification.

When the sequence generator finds a statement that passes all the viewspec requirements, it returns the statement to the formatter and waits to be called again for the next statement in the sequence.

Formatter 4d1e4c4

The formatter arranges text passed to it by the sequence generator (described below) in the style specified by the user. The formatter observes viewspecs such as line truncation, length and indenting; it also formats the text in accord with the requirements of the output device.

The formatter works by calling the sequence generator, formatting the text returned, then repeating this process until the sequence generator decides that the sequence has been exhausted or the formatter has filled the desired area (e.g., the display).

User Filters 4d1e4c5

The user filter program may be either a content analysis pattern or a more complex L10 program.

Content Analysis Patterns

Content analysis patterns describe characteristics that a statement must have to be included in the sequence being generated. For example, a content analysis pattern may stipulate that a statement must contain a particular phrase, or that it must have been written since a particular date. In general, content analysis patterns may use any of the pattern matching facilities permitted in L10 FIND statements.

Content analysis patterns cannot affect the format of a statement, nor can they initiate editing operations on a file. They can only determine whether a statement should be viewed at all.

Nevertheless¤@ content analysis filters provide a powerful tool for user control of the portrayal of a series of statements. They are the most frequently used, and easily written, of the user programs. However, if one wishes to change the format of a statement, or to modify the file as it is displayed, he must use a user written L1O program.

User Written L1O Programs

A user written program may be given control by the sequence generator in exactly the same fashion that a content analysis program is initiated. However, in addition to pattern matching, it may change the format of a statement being displayed and may modify the statement itself (as well as other statements in the file).

A user written program invoked by the sequence generator has several limitations. It can manipulate only one file and it can look at statements only in the order in which they are presented by the sequence generator. In particular, it cannot back up and re-examine previous statements, nor can it skip ahead to other parts of the file. A user-written sequence generator must be provided when one needs to overcome these restrictions.

User-Written Sequence Generators 4d1e4c6

A user may provide his own sequence generator to be used in lieu of the regular NLS sequence generator. Such a program may call the normal NLS sequence generator, as well as content analysis filters and user-written L1O programs. It may even call other user-written sequence generators.

This technique provides the most powerful means for a user to reformat (and even create) multiple files and to affect their portrayal. However, since writing them requires a detailed knowledge of the entire NLS program, the practice is limited to experienced NLS programmers.

Examples Of Content Analysis Patterns And L10 User Programs 4d1e4d

The user-written filters may be imposed by an NLS subsystem accessed by the command "Goto Programs". 4d1e4d1

These NLS commands are used to compile, institute (or link the compiled user program into the user's copy of the running NLS system), and execute User Programs and filters. They are described in detail in the L10 Primer. (9246,). 4d1e4d2

Examples of simple content analysis patterns and L10 analyzer-formatter user programs follow. 4d1e4d3

Examples of Simple Content Analysis Patterns 4d1e4d4

BEFORE (25-JAN-72 12:00);

This pattern will match those statements created or modified (whichever happened most recently) before noon on 25 January 1972.

ID = HGL OR ID = MFA;

This pattern will match all statements created or modified (whichever happened most recently) by users with the identifiers "HGL" or "MFA".

D 2$LD / [/"CA" / "Content Analyzer"/];

This pattern will match any of three types of statements: those beginning with a numerical digit followed by two characters which may be either letters or digits, and statements with either the patterns "CA" or "Content Analyzer" anywhere in the statement.

Note the use of the brackets to permit an

unanchored search -- a search for a pattern anywhere in the statement. Note also the use of the slash for alternations.

```
[(2L (SP/TRUE) /2D) D '- 4D];
```

This pattern will match characters in the form of phone numbers anywhere in a statement. Numbers matched may have a two digit alphabetic exchange followed by an optional space (note the use of the TRUE construction to accomplish this) or a numerical exchange.

Examples include YU 4-1234, YU4-1234, and 984-1234.

Examples of Analyzer-Formatter Programs 4d1e4d5

The following are examples of user analyzer-formatter programs which selectively edit statements in an NLS file on the basis of text searched for by the pattern matching capabilities. Examples of more sophisticated user programs such as sort keys and user sequence generator programs will be presented in a later supplement with a description of NLS routines easily accessed by users.

Example 1--

```
PROGRAM outname % removes statement names --
del= () --%
DECLARE TEXT POINTER sf, paf, pae;
(outname)PROCEDURE;
IF FIND ↑sf $NP '( ↑paf [')] ↑pae THEN
BEGIN
ST sf ← pae SE(sf);
RETURN(TRUE);
END
ELSE RETURN(FALSE);
END.
FINISH
```

This program removes any parenthesized expression whose opening parenthesis

corresponds to the first printed character of an NLS statement.

Example 2--

```
PROGRAM changed
(changed)PROCEDURE;
LOCAL TEXT POINTER f, e;
FIND ↑f SE(f) ↑e;
IF FIND SINCE (25-JAN-72 12:00) THEN
BEGIN
ST f ← "/CHANGED/", f e;
RETURN(TRUE);
END
ELSE RETURN(FALSE);
END.
FINISH
```

This program checks to see if a statement was written after a certain date. If it was, the string "/CHANGED/" will be put at the front of the statement.

Software Engineering Augmentation Systems (SEAS) 4d2

Introduction 4d2a

Of all of the special application areas where our augmentation tools could reasonably be applied for testing and evaluation, that of the software engineer has from the beginning been our prime candidate. We took a significant step in this direction in 1968 when we developed MOL940, a special, higher-level language, and applied it to all of our NLS programming. MOL940 allowed our software engineers to use the special features of NLS for supporting the composition, studying, and modification of our source code and its documentation. The result was a significant step in augmenting their capability. 4d2a1

In this past contract period, we have taken several steps to further augment the software engineer -- in fact, we have coined the acronym SEAS (for Software Engineer Augmentation System) to give specific system orientation towards the end of developing a full and balanced set of tools, techniques, methods, principles, etc. for augmenting software engineers. The developments described below are

part of an accelerating activity -- an important part of our near-future plans in the next contract period involve a greater level of activity here. 4d2a2

The SEAS developments summarized below are described in more detail in the following sections: 4d2a3

With the change from our XDS-940 to the PDP-10, we upgraded our compiler compiler to a more flexible Tree-Meta Compiler,--,4d2e) and our system-programming language to the more powerful, less machine-dependept L10; both developments added to the SEAS tool kit. 4d2a3a

We adopted new standards for documentation, and developed several system-measurement sub-systems, see--,7b3). 4d2a3b

During the last year, we developed a source-code debugging system for L10, working from NLS see--,4d2b). Source Level Debugging not only will be useful to us for the remaining period of our L10 usage, but also it serves as a prototype of an approach which will be applicable for others who can utilize an NLS-based SEAS for software engineers that use another language such as PL1, COBOL, FORTRAN, or even an assembly language. 4d2a3c

During the last year, we also began development work on the next stage of compiler compiler, and an advanced, modular, system-programming language (MPS) see--,4d2f). which won't be finished until halfway through the next contract period -- but which will provide a significant step forward for SEAS. We will use them to implement the succeeding stages of NLS evolution, and they will also provide the base for the intensive exploratory developments of our central, advanced SEAS experiments. 4d2a3d

Source Level Debugging 4d2b

By making minor changes to the TENEX Dynamic Debugging Technique system, DDT, and to the ARC L10 programming language compiler, and by providing a fairly simple debugging submode accessible through NLS, NLS-DDT, ARC software engineers have provided themselves with a primitive but effective source level debugging and (procedural level) incremental compilation system. 4d2b1

This system was developed as a user program and is currently functional only in TNLS. It will soon be expanded to DNLS as well. Documentation of the commands in the system may be found in (Journal, 8334,). 4d2b1a

The NLS-DDT system provides an easier way to examine individual cells and L10 data structures, such as records, fields, strings, and call stack frames, than is available in the current TENEX DDT. 4d2b2

procedures which are compiled in the User Program submode may replace procedures in a running system during a debugging session without the necessity of either patching in machine language code, as in the TENEX DDT, or loading an entirely new system, a slow process for a large, multi-file program such as NLS. Symbol definition is resolved with the rest of the running code. Such procedures may also be inserted into the program. 4d2b3

The breakpointing features of TENEX DDT are provided as well as a conditional breakpointing capability. 4d2b4

The command language is less obscure than that of TENEX-DDT and is more consistent with other commands in the NLS environment. 4d2b5

System Measurement 4d2c

The designers of a continually evolving system must be able to measure the effectiveness of modifications introduced into the whole system. They must be able to quantitatively and qualitatively measure the effect of a change on the command use of individual users and on the whole system response. Analyses of these measurements indicate the need for modification in training techniques and for further changes. 4d2c1

NLS can measure its own activity in various ways. Each of these measurement techniques was added to NLS at different times and in response to different questions the system programmers were asking about system activity. 4d2c2

These primitives will be expanded to be used with the more formal measurement and evaluation goals of SEAS. 4d2c2a

measuring the elapsed time between two instructions. 4d2c3

This is the crudest measurement facility; the only user interface is through the PDP-10's DDT subsystem. Given two addresses and a count, the elapsed job time between executing the two instructions will be accumulated the number of times specified by the count. Then the figures are reinitialized and the time reaccumulated. 4d2c3a

Measuring the time required by various types of NLS commands. 4d2c4

The real and job times required to execute various types of NLS commands can be collected at regular intervals and saved on a file. The queue number, number of reserved pages, number of page faults, and working set size, averaged over the interval, are also recorded. This file must then be processed by a separate program to interpret and format the results. 4d2c4a

Four basic types of statistics are collected -- information about text editing commands, about structure editing commands, about the time NLS requires to respond to a single character, and about the lag between the time the user types a character and the time NLS receives it. 4d2c4b

Monitor measurements 4d2c5

Several monitor calls have been added to help in the measurement of our system. For example, one of these collects information from NLS about the real and execution time required for each interaction with the user. 4d2c5a

The measurement facility for the entire timesharing system, Superwatch, is described below. (See --,7b3) 4d2c6

Source Code Documentation Standards 4d2d

Several programmers continually modify the 150,000 computer words of NLS code. In such a large system it is essential that code be clearly documented to permit anyone to fix bugs and make additions to the system as flexibly and easily as possible. Well documented source code, viewed using the linking and level-clipping features of NLS, provides an immediate overview of the system and an important tool to the augmented software engineer. 4d2d1

The lack of proper documentation clearly becomes untenable in a bootstrap community with many widely dispersed people collaborating on the same system. 4d2d1a

Thus, in the development of a software engineering system design discipline, standards and methods for documentation must exist. Toward this end, several steps were taken in the last contract period. 4d2d2

Standards for documentation and coding were proposed in (Journal,8573,), (Journal,8637,), and (Journal,8643,). They have been used in cleaning up several NLS source code files. This clean up is continuing. 4d2d2a

A program for developing a linked cross index has been in use for several months. 4d2d2b

Languages 4d2e

Introduction 4d2e1

ARC currently makes use of two primary languages created at the center in its NLS system development: the L10 programming language, which is used to write NLS programs, and the Tree-Meta compiler-compiler system, which is used to generate compilers for L10, have been used to bootstrap compilers onto different computers, and have been used to generate the first compiler for the Modular Programming Language (MPL). 4d2e1a

Additionally, Tree-Meta has been used to develop an interpreter for the output processor directive language. 4d2e1a1

In collaboration with several people at the Xerox Palo Alto Research Center, work has begun on a Modular Programming System (MPS) and a Modular Programming Language (MPL) that will replace the current languages and in which the NLS system will be redesigned and rewritten for greater efficiency and flexibility. 4d2e1b

L10 4d2e2

NLS on the PDP-10 is written in the L10 programming language, an ALGOL-like language that has some high

level special purpose features for string analysis and manipulation and for interacting with NLS users. 4d2e2a

The June 1971 report (8277,) describes the process of transferring from the XDS-940 languages and compilers to the PDP-10. An L10 primer (9246,) describes many of the features of the language for inexperienced programmers wishing to make use of the User Program facility. A complete presentation of the language is also available in a terser form for experienced programmers. 4d2e2b

Tree-Meta 4d2e3

Tree Meta is a metacompiler system for context-free languages developed at ARC. The parsing statements of the metalanguage resemble Backus-Naur Form with embedded tree-building directives. Unparsing rules include extensive tree-scanning and code-generation constructions. All compilers produced by the system are single pass compilers that produce loadable binary files. 4d2e3a

A metacompiler, in the most general sense of the term, is a program that reads a metalanguage program as input and translates that program into a set of instructions. If the input program is a complete description of a formal language, the result of the translation is a compiler for the language. 4d2e3a1

Tree Meta is built to deal with a specific set of languages and an even more specific set of users. There is no attempt to design universal languages, or machine independent languages, or to achieve any of the other goals of many compiler-compiler systems. 4d2e3b

In the past contract period Tree Meta was useful in bootstrapping from the old XDS-940 to the new PDP-10. Currently it is being used to create the first MPL compiler. 4d2e3c

A version of Tree Meta was discussed in an appendix to the Rome Report of April 1968 (9697,). Since that time, the syntax has been expanded and the system made more flexible. A new Tree-Meta report 10869,) includes a formal description of the Tree Meta language taken from a longer Tree Meta report being completed. 4d2e3d

Modular Programming 4d2f

Goals 4d2f1

The Modular Programming System (MPS) is a set of tools for the development and continued evolution of large software systems in an interactive environment. All such large software systems share certain characteristics: 4d2f1a

(a1) they are the work of a group of people whose membership will change over time; 4d2f1a1

(a2) they are necessarily constructed from a number of separately developed programs; 4d2f1a2

(a3) they evolve and grow throughout their lifetimes (and there is evidence that they also "age" (10481). 4d2f1a3

The MPS project aims to decrease the effort required to build and evolve such systems and to increase the reliability of the resultant products. As a specific test of its capabilities, MPS will be used in the rewriting and restructuring of the NLS system developed at Stanford Research Institute. 4d2f1b

Desirable Characteristics 4d2f2

Points a1, a2, a3 are axiomatic statements about the dynamics of all large software systems. The following discussion uses these and a few other axioms to establish desirable characteristics for MPS. They are intended only to lend plausibility to the set of capabilities which the MPS project is investigating. Furthermore, the "logical conclusions" only represent design choices to satisfy the axioms; other choices could certainly be made which would not be inconsistent with the axiom set, but that is another research project. Hopefully there is a minimum of hidden meaning in the following discussion: each axiom and consequence is intended to be taken strictly at face value. 4d2f2a

We first add two more axioms to the above set: 4d2f2b

(a4) Large software systems must be able to take advantage of available hardware for efficiency. 4d2f2b1

(a5) Program bugs are not known before they occur. 4d2f2b2

(a4a) a1-a4 imply that software components, nereafter called modules, should be separately compilable and debuggable. Therefore there must be a way of linking or binding separate components together to provide an environment (data and programs) within which a module can be debugged. 4d2f2c

(a6) In an interactive programming environment, users must be able to develop and use debugging tools applicable to programs in the same programming system (6035,) (10478,). 4d2f2d

a4a, a5, and a6 then imply that 4d2f2e

(a6a) the environment of a program must be dynamically alterable; 4d2f2e1

(a6b) a program should not have to be altered when its environment changes in ways which do not affect the semantic intent of the program -- this is called programming generality. 4d2f2e2

(a3a) a3 suggests that a desirable characteristic for tools for building large systems should be that the energy to change part of the system should be more a function of the complexity of the change than of the size of the system. 4d2f2f

(a3b) A new system always has parts which are functionally similar to previously developed systems. The new system may therefore be regarded as a change (though perhaps substantial) to an older system. a3a then points out the necessity for being able to reuse components which have been made reliable through usage. This increases the initial reliability of the new system and decreases its cost. 4d2f2g

(a3c) One way of constructing useful components is to build them from combinations of already existing modules (a3b). Hence there must be a way of bundling useful configurations together as seemingly atomic modules so they can be readily reused. 4d2f2n

MPS Capabilities 4d2f3

To satisfy these objectives, MPS has concentrated on providing the following capabilities: 4d2f3a

Control mechanisms which enable modules to be linked together with a minimum of built in assumptions about how each interprets control transfer over the link between them. 4d2f3a1

Simple function call and return mechanisms alone do not satisfy this requirement.

Data definition facilities that: 4d2f3a2

clarify the specification of the data structures which, together with control, completely specify the interfaces between modules;

are potentially economical in space and accessing speed without being dependent on a particular machine;

are an aid in developing and describing program components and the structure of algorithms.

Facilities for dynamically binding the virtual objects required by a module for execution to real objects. 4d2f3a3

For example, for binding a procedure call to a real procedure, a "typed" pointer to a data structure of the correct type, etc. The set of bindings for a module's virtual objects at a given moment comprises the environment for that module.

Complete accessibility to the MPS "virtual machine" (which is a set of primitive MPS programs) and to MPS programs as data structures. 4d2f3a4

This enables debugging and measuring tools to be built as standard MPS programs and along with dynamic binding allows such tools to be brought to bear on MPS programs whenever necessary.

The ability to bundle a configuration of data and program modules together as a module which may be saved for later use just as a simple, atomic module. 4d2f3a5

This allows systems to be partly initialized by partially executing them and then bundling them up for later use with the initialization computations factored out;

It also allows a configuration that has exhibited a bug to be saved away for later perusal with the state as it was when the bug was discovered;

Lastly, it allows standard modules to be built by configuring them from other modules in the spirit of using already available components whenever possible and provides some logical completeness to the system.

INTERNAL ORGANIZATION 4e

During the past year, several ARC organizational arrangements were introduced, centering, in the early part of the period, mainly on line activity structure and associated roles. 4e1

The creation of pusher (task leader) roles for tasks and coordination roles for system architecture, methodology, and personnel resources placed the responsibility for these efforts more directly on selected individuals. 4e1a

pusher roles were carried out in the framework of the developing Baseline Management System. Coordinating roles were also carried out in this environment. The techniques for performing these roles still leave much to be desired. 4e1a1

Our plans to record task requirements and designs will aid this process. 4e1a1a

In the Fall of 1971, we set up a four-man Executive Management Committee (EMC) to carry out many of the day-to-day operating management tasks. Membership was later changed to three. 4e1b

The EMC has documented its meetings through Journal entries as they occurred. 4e1b1

PODAC is to deal with ARC peoples' beliefs, interests, and feelings, helping people and the organization to deal with the goals and line activities that result. 4e1b2

During the past few months, a new, more broad overall organizational structure has been in the process of formation. 4e1c

This consists of three main activities that have been set up to cover our framework and goal setting, line operation, and personal and organizational development needs. 4e1c1

These activities are called: FRAMAC, LINAC, and PODAC. 4e1c2

FRAMAC is to discuss and define the ARC framework and set long-range goals and plans. 4e1c2a

LINAC is to carry out activities within the framework that move us toward the goals, with more detailed, shorter-range plan formulation. 4e1c2b

PODAC is to deal with ARC peoples' beliefs, interests, and feelings, helping people and the organization to deal with the goals and line activities that result. 4e1c2c

These are described in more detail below and in documents (10331,), (10034,), and (8651,) respectively. 4e1c2d

FRAMAC 4e2

We have launched an activity within ARC called our Framework Activity (FRAMAC). 4e2a

FRAMAC's goals and general method of approach are: 4e2b

To provide a continuing, purposefully run forum, for developing the framework of concepts, strategies, principles, and goals within which we will pursue our planning, promoting, growing, LINAC and PODAC activities, and interaction with the world. We are holding a regular sequence of meetings, where dialog is expected. Records are kept and Journalized. A coherent, explicitly developed Framework Section of the Handbook will ensue. 4e2b1

The first meetings' notes are recorded in (10458,), (10459,), and (10553,). 4e2b1a

Our First Stage (starting May 1972 and lasting several months) includes: 4e2b2

a) Piecing together and bring about a general understanding of Dr. Engelbart's personal framework, the history that brought us to where we now are, and the current state of our implicit framework (i.e. the practices, principles, goals, etc. that we can see have affected our current state and direction). 4e2b2a

b) Bringing each of our FRAMAC participants to understand reasonably well where each of the others stands on what we consider to be the important facets of the framework, in terms not only of degree of his understanding, but also of the degree and nature of his interest, beliefs, and attitudes. 4e2b2b

Our Second Stage will include: 4e2b3

A continuing process of framework analysis and

development. The objective is to continually evolve toward a "most useful framework," one that is kept complete and updated as part of our Handbook, and that is referenced constantly in our planning, designing, evaluating, and teaching. 4e2b3a

We plan that in this stage we would judiciously integrate concepts, considerations, viewpoints, and analyses of others, via an organization and process yet to be decided upon. During the Stage 1 process, Dr. Engelbart will further develop parts of his framework and will describe those parts that bear upon the process of further ARC Framework development. 4e2b3b

We plan on an approach here that is much as if we were running a graduate seminar to impart where Dr. Engelbart is in his thinking. An unbroken series of individual presentations (lecture model) won't accomplish what we want. We expect to have both prepared and extemporaneous presentations, but in limited cuts and modules from Dr. Engelbart's and others' frameworks, interspersed with multi-way group dialog sessions each of whose content affects succeeding presentations. We don't know where most of the participants are in their thinking now, with respect to understanding most of the issues involved, nor what kind of presentation it would take to produce a given change in understanding on any given issue. 4e2c

We speak of developing a "general understanding" of our framework (which may involve a lot of work); but there also is the matter of the distribution among the participants in the nature and degree of their "beliefs and attitudes" (B&A) about the various facets of the framework. It is important for Dr. Engelbart at least to know what this B&A distribution is; and it may prove important to the succeeding FRAMAC stages to work at bringing about a closer grouping of ARC peoples' B&A relative to certain issues. We expect that we will want to deal with this, but how much energy to spend, and what part within FRAMAC and what part in PODAC, will have to be decided as we progress. 4e2c1

About the initial composition of our FRAMAC group: 4e2d

We had been visualizing a small FRAMAC group, considering the type of dialog we hope for. But when we reviewed our LINAC planning-team composition, we decided that there is a such strong interaction between our current planning

exercise and our Framework that we couldn't seem to find a logical way to cut the group membership smaller. The initial FRAMAC group numbers nineteen ARC people plus two other SRI management people. 4e2d1

LINAC 4e3

We have launched an activity within ARC called our Line Activity (LINAC). 4e3a

LINAC serves several basic needs: 4e3a1

Modularizing our way of doing things -- something that the size and complexity of our activities require. 4e3a1a

Establishing interdependence relationships that will give us valuable experience for the future problems of managing a considerably larger and more varied activity within an increasingly complex operational and technical environment. 4e3a1b

Establishing the activity framework within which we can pursue our new-contract commitments to ARPA (as per our proposal of 29 July 1971 -- 7404,) 4e3a1c

In LINAC's organization, our external projects are the driving forces -- where a project is an explicit activity involving resource interchange with outside organizations. The other specific activities within ARC are to serve the projects' goals, and will have all of their resources allocated, along a contracting chain, from the projects. 4e3a2

Along with this (internal) contracting system will come specific development and application of conventions, procedures and aids for handling estimates, resource allocations, budgets, reserves, accounting and resource-control measures as required to operate the organization. 4e3a2a

We expect that many of our internal activities will emerge from multi-party negotiations and proceed under contracts involving several buyers. 4e3a2b

Some of our activities will be funded by what amounts to a taxation upon all or some of the projects. Such

taxation measures will be established and monitored with due representation by the concerned parties. 4e3a2c

ARC planning and task activities are currently conducted in the following LINAC organization: 4e3b

OPERATIONS 4e3b1

- Administration
- Computer Service Operations - Hardware
- Computer Service Operations - Software
- Computer Service Operations - Operators
- People Service Operations
- User Interface

DEVELOPMENT THRUSTS 4e3b2

- Development Coordination
- Delivery and Marketing
- Dialog Support System (DSS)
- Documentation Production and Support System (DPCS)
- Baseline Record System (BRS)
- System Developers Handbook System (SDHS)
- Software Engineering Augmentation System (SEAS)
- General Development (not included in above thrusts)

PROJECTS 4e3b3

- ARPA/RADC Project: Team Augmentation Portion
  - Administration
- ARPA/RADC Project: Network Information Center Portion (NIC)
  - Administration
  - Computer Service Operations
  - People Service Operations
  - Net Interface (Station Agent and Net participation)
  - NIC Development
- ARPA/RADC Project: Mini-Console
  - Administration
  - System Development
- ARPA/RADC Project: MPS Cooperation (Xerox)
  - Administration
  - Modular Programming System Development (MPS)
- ONR Project: System Developer's Intelligence System (SDIS)
  - Administration

RADC Project: Baseline Management System Development
Support
Administration

SRI OVERHEAD ACTIVITIES 4e3b4

During the early stages of the new LINAC, the following actions are taking place: 4e3c

Each of the main activities is developing the framework of a plan, with a reasonable amount of informal intercommunication and coordination between plans. 4e3c1

Eight people who carry key ARC planning roles will meet regularly to serve as a "Planning and Executive-Review committee" (PERC). 4e3c2

One important function for PERC during this time will be to develop recommendations for refinements to the LINAC system of roles and processes. 4e3c2a

Another function will be to participate in and review the operational decisions that must be made to coordinate and manage the efforts of the projects and developmental thrusts. 4e3c2b

Within the three parallel pushes of FRAMAC, LINAC, and PODAC, our persistent emphasis will be toward "coordinated-system" aspects of both our way of working and of the augmentation system(s) we develop. 4e3d

PODAC 4e4

In January 1972, ARC established a regular channel for Personal and Organzational Development named PODAC. Our planning for PODAC was integrated with planning for LINAC and FRAMAC discussed above. 4e4a

Establishment of PODAC arose from the conviction that we, who tell the world that we are learning how to show other teams how to pursue goals more effectively, must constantly examine ourselves (the "example" that we work with), both as an organizaton and as individuals, to understand how we are doing, and how we can improve. 4e4a1

We are convinced that unless we have a strong, constant,

and pervasive attitude that we want to keep developing ourselves, and unless we consciously keep trying to do so, then we are fooling ourselves about seriously pioneering this augmentation system development. 4e4ala

To work on this, we need a flow of information having to do with goals, attitudes, ambitions and feelings as they relate to the common pursuits, and purposeful discussion about strengths, weaknesses, and means for improvement. 4e4alb

To establish PODAC, we divided the staff into four groups of eight or nine people each. 4e4b

The groups, called POD's, are balanced in age, sex, professional training, length of association with ARC, work roles, etc. 4e4b1

POD's are named Cedar, Fir, Oak, and Redwood. 4e4b2

Each group meets weekly for two hours. 4e4b3

Each such group appoints its own representative to a central committee, PODCOM, that helps to co-ordinate and guide the PODAC. 4e4b4

PODAC does not exist to vote on what ARC will do. PODAC has no line-management responsibilities or authority. It is "orthogonal" to the management structure that commits resources, sets targets, hires, reviews, and is held accountable. 4e4c

Instead, it provides an organized mechanism for interactions among all parties toward affecting the understanding, beliefs, and attitudes of each other, as a means of affecting the decisions and actions within ARC, toward what each thinks is the best set of goals, organization, products, behavior. 4e4c1

It is a forum for the expression of concerns, beliefs, ideas, feelings, and dissension existing within any person or group in ARC about the way things are being done (or not being done), about our goals, etc. 4e4c1a

It is a way to keep everyone informed about the problems and opportunities facing ARC and its people and its goals. 4e4c1b

SRI-ARC 6 JUNE 1972 13041
Team Augmentation
Internal Organization

PODAC has been active for three months at the end of this contract period. It is not yet easy to evaluate our accomplishments. 4e4d

Meetings of the groups described varied considerably in content. 4e4d1

On one hand many members feel that people now communicate somewhat more easily among themselves within the POD's and feel that they had some fruitful discussions of the goals and strategy of our research and of personal effectiveness at work. 4e4d1a

On the other hand many some people have felt indifferent, hostile, or anxious when confronted with the mandatory but undefined participation, and have withdrawn or participate only very passively. 4e4d1b

Very little agreement on large issues or other action has yet resulted. 4e4d1c

PODAC has invited speakers on Organizational and Personal Development, instituted a small library in the field, instituted augmented procedures for cataloging the library, and formed several special interest subgroups. 4e4d2

REFERENCES 4f

(5139,)D.C. Engelbart and Staff of ARC (SRI-ARC), Computer Augmented Management-System Research and Development of Augmentation Facility--Final Report Augmentation Research Center, Stanfor Research Institute, Menlo Park, Califorinia 94025, RADC-TR-70-82, AD 709,211. April 1970. 268p. 4f1

(6035,) R. A. Krutar. Conversational Systems Programming - Or Program Plagiarism Made Easy. Computer Science Department, Carnegie-Mellon University, Pittsburgh, Pennsylvania 15213. 3 March 1971. 23p. 4f2

(6912,)Walter L. Bass (SRI-ARC). Output Processor Brief User Guide. Augmentation Research Center, Menlo Park, California 94025, 27 April 1971. 8p. 4f3

(7404,) Proposal for Research [to RADC/ARPA ISU 71-94]. Network Information Center and Augmentation System Development. SRI No. ISU 71-94. Augmentation Research Center, Stanford Research

Institute, Menlo Park, California 94025. 29 July 1971. Separately paged. (Entered in SRI-ARC Journal 18 October 1971.) 4f4

(7470,) Network Information Center, Augmentation Research Center, Stanford Research Institute, Menlo Park, California 94025. TNLS User Guide: Preface, Syntax and Contents. 1 September 1971. Separately paged. 4f5

(7637,) NIC Journal System User Guide, Section 2, Journal System. Augmentation Research Center, Network Information Center, Stanford Research Institute, Menlo Park, California 94025. 1 October 1971. 13p. 4f6

(7638,) NIC Journal System User Guide, Section 3, Identification System. Augmentation Research Center, Network Information Center, Stanford Research Institute, Menlo Park, California 94025. 1 October 1971. 16p. 4f7

(7639,) NIC Journal System User Guide, Section 4, Number System. Augmentation Research Center, Network Information Center, Stanford Research Institute, Menlo Prk, California 94025. 1 October 1971. 5p. 4f8

(8277,) D. C. Engelbart (SRI-ARC). Network Information Center and Computer Augmented Team Interaction, Interim Technical Report. Augmentation Research Center, Stanford Research Institute, Menlo Park, California 94025. Rome Air Development Center, ARPA. RADC-TR-71-175, AD 737 131. 30 June 1971. 104p. 4f9

(8651,) D. C. Engelbart (SRI-ARC). To Launch PODAC. Augmentation Research Center, Stanford Research Institute, Menlo Park, California 94025. 25 January 1972. 6p. 4f10

(9241,) Harvey G. Lehtman (SRI-ARC). DEX-2 Proposed Design. Augmentation Research Center, Stanford Research Institute, Menlo Park, California 94025, 22 February 1972. 24p. 4f11

(9246,) Augmentation Research Center, Stanford Research Institute, Menlo Park, California 94025. L-10 Programming Guide (a user guide). 4 April 1972. 100p. 4f12

(9697,) D. C. Engelbart, W. K. English, J. F. Rulifson (SRI-ARC). Development of a Multidisplay, Time-Shared Computer Facility and Computer-Augmented Management-System Research. Augmentation Research Center, Stanford Research Institute, Menlo Park,

California 94025. Rome Air Development Center, ARPA. AD 843 577. April 1968. 180p. 4f13

(9934,) Augmentation Research Center, Stanford Research Institute, Menlo Park, California 94025. Deferred Execution (DEX) User Guide. 16 June 1972. 66p. 4f14

(10034,) D.C. Engelbart (SRI-ARC). To Launch LINAC. Augmentation Research Center, Stanford Research Institute, Menlo Park, California 94025. 7 April 1972. 6p. 4f15

(11076,)SRI-ARC,OUTPUT PROCESSOR USER' GUIDE -- Introduction,Augmentation Research Center, Stanford Research Institute, Menlo Park, California 94025. 6 September 1972. 5p. 4f16

(10331,) D. C. Engelbart (SRI-ARC). To Launch FRAMAC. Augmentation Research Center, Stanford Research Institute, Menlo Park, California 94025. 4 May 1972. 3p. 4f17

(10457,) James C. Norton (SRI-ARC). Initial FRAMAC Meeting Notes. Augmentation Research Center, Stanford Research Institute, Menlo Park, California 94025. 23 May 1972. 14p. 4f18

(10459,) James C. Norton (SRI-ARC). Second FRAMAC Meeting Notes - Centering on the Concept of Frameworks. Augmentation Research Center, Stanford Research Institute, Menlo Park, California 94025. 1 June 1972. 39p. 4f19

(10478,) James G. Mitchell, A. J. Perlis, H. Van Zoeren. LCC: A Language for Conversational Computing. In: J. Reinfelds, M. Klerer, Eds. Interactive Systems for Experimental Applied Mathematics. Academic Press, New York. 1968. p.203-214. 4f20

(10479,) James G. Mitchell. LCC. In: Computer Science Research Review, Carnegie-Mellon University. 1969. p.21-29. 4f21

(10481,) L. A. Belady, M. M. Lehman. Programming System Dynamics -- Or the Meta-Dynamics of Systems in Maintenance and Growth. IBM Research Document RC 3546. T. J. Watson Research Center, International Business Machines Corporation, Yorktown Heights, New York. 17 September 1971. 4f22

(10553,) Richard W. Watson (SRI-ARC). FRAMAC Notes, May 19 [1972]. Augmentation Research Center, Stanford Research Institute, Menlo Park, California 94025. 19 May 1972. 2p. 4f23

(10703,) Augmentation Research Center, Stanford Research Institute, Menlo Park, California 94025. DNLS Preliminary Reference Guide. 21 June 1972. Separately paged. 4f24

(10869,) D. I. Andrews, H. G. Lehtman, W. H. Paxton (SRI-ARC). Tree Meta - A Metacompiler for the Augmentation Research Center. In process. Unpaged. 4f25

SRI-ARC 8 JUNE 1972 13041
Network Information Center
Development and Operations
Introduction

NETWORK INFORMATION CENTER

by Richard W Watson, Jeanne B North, James E White,
John T Melvin, Walter L Bass, James C Norton,
Cindy Page, Dirk H van Nouhuys, and the Staff of ARC 5

INTRODUCTION 5a

The ARPA Computer Network (ARPANET) has been established to provide both a new experimental type of communication facility and a base for resource sharing. 5a1

The ARPANET community can be viewed as a collection of resources, people, hardware, software, data, and special services which can be brought together for short or long periods in different configurations to work cooperatively on a given problem or task. 5a2

In this context the development of the ARPANET can be viewed as a multileveled experiment in learning how to bring together and make available these distributed resources. 5a2a

At the lowest level are the problems of creating a basic communication facility which allows different types and configurations of computer hardware to communicate. 5a2b

At intermediate levels are the developments of protocols which allow classes of computer programs to communicate with each other and permit data to be shared. 5a2c

At higher levels still are the processes which assist people to find the geographically distributed facilities they need to solve or study a problem and which allow distributed people to work together effectively. 5a2d

The Network Information Center (NIC) is one part of the ARPANET experiment interested in the higher levels of problems. A service such as the NIC helps to create and sustain the sense of community needed in an experiment such as that of the ARPANET. The NIC is more than a classical information center, as that term has come to be used, in that it provides a wider range of services than just bibliographic and "library" type services. 5a3

The Network Information Center (NIC) is an experiment in setting up and running a general purpose information service serving the ARPANET community (both those individuals and groups with direct

access to the network, and those associated with work going on in the network but without direct access) with both online and offline services. The services offered and under development by the NIC have as their initial basic objectives: 5a4

1) To help people with problems find the resources -- people, systems, and information -- available withn THE NETWORK COMMUNITY WHICH MEET THEIR NEEDS. 5a4a

2) To help members of a geographically distributed group collaborate with each other. 5a4b

THE NIC PUBLIC 5b

To provide reliable, useful effective information services to meet the basic needs of a growing, diverse ARPANET community will offer considerable challenge. 5b1

One of the problems in the design of an information service is to determine the main classes of clientele which exist for this service and to determine their needs. 5b2

The initial clientele for NIC services are those people developing and building the network. The next group is composed of those people whose research and development interests are intimately connected with network resources or who would be experimental users of various network resources. After this initial period the classes of clientele will grow, as the network becomes a well shaken down operational entity, to include a wide range of people who will use the network or be interested in its development. 5b2a

Our initial analysis showed us that there were four main needs which the NIC could attempt to meet, Reference and General Network Information, Collaboration Support, Document Handling and Creation, and Training. Although training programs must eventually exist for all services available on the network, our initial emphasis is training in the use of NIC services. 5b2b

Some users of the Network Information Center's services may be: 5b3

- Students
- Researchers
  - university
  - industry
  - government
- System Developers
  - university
  - industry
  - government
- Teachers
- Managers
  - university
  - industry
  - government
- Computer Center Directors
- Libraries and Other Information Services
- The General Public
- The Media

PRESENT NIC SERVICES 5c

The initial NIC services now available to meet the above goals and present clientele are the following: 5c1

Online: 5c1a

(1) Access to the typewriter version (TNLS) of the Augmentation Research Center's Online System (NLS) for communique creation, access, linking between users, and for experimental use for any other information storage and manipulation purpose suitable for NLS and useful to Network participants. 5c1a1

(2) Access to Journal, Number, and Identification Systems which allow messages and documents to be transmitted to Network participants. 5c1a2

(a) Documents or messages entered in the Journal System are maintained online for later viewing via NLS. 5c1a2a

(b) Documents are now distributed by: 5c1a2b

i) placing the message or a link to the document in the receiver's "initial file". 5c1a2b1

ii) sending hardcopy through the U.S. mail. 5c1a2b2

Documents will shortly be distributed through the Network when sites have implemented the appropriate File Transfer Protocols 5c1a2c

(c) A unique number is assigned each entry at the time of submission. Numbers can also be preassigned to allow related documents to be interlinked at the time of their preparation. 5c1a2d

(d) A catalog entry is prepared at the time of submission and later this entry is used to update a catalog kept both online and in hardcopy form. 5c1a2e

(e) Special interest groups can be created to facilitate indicating to the system particular distribution lists for dialog items. Dialog items can be placed in subcollections associated with the dialog groups for special index production. 5c1a2f

(3) Access to a number of online information bases through a special Locator file using NLS link mechanisms. 5c1a3

(a) Links to the NIC functional documents, including the printed catalog of the NIC document collection, the ARPA Network Resource Notebook, NIC user documentation, a Directory of Network Participants, and Network Protocols 5c1a3a

(b) Links to other files created by sites with information of potential Network-wide interest. 5c1a3b

Offline: 5c1b

(1) A Network Information Center Station set up at each site with: 5c1b1

(a) A Station Agent to aid use of the NIC 5c1b1a

(b) A Liaison to provide technical information about his site. 5c1b1b

(c) A Station Collection containing a subcollection of documents of interest to Network participants. 5c1b1c

(2) Techniques for gathering, producing and maintaining NIC Functional Documents such as: 5c1b2

(a) Current Catalog of the NIC Collection 5c1b2a

(b) ARPA Network Resource Notebook 5c1b2b

(c) Directory of Network Participants 5c1b2c

(d) NIC User Guide 5c1b2d

(3) Support of Network dialog existing in hardcopy through duplication, distribution, and cataloging. 5c1b3

(4) General Network referral and handling of document requests 5c1b4

(5) Building of a collection of documents potentially valuable to the Network Community. Initial concentration has been on obtaining documents of possible value to the Network builders. 5c1b5

(6) Crude selective distribution to Station Collections. 5c1b6

(7) Training in use of NIC services and facilities. 5c1b7

In the sections to follow each of the above services and its supporting technology and organization will be discussed in more detail. 5c1c

RELATION OF THE NETWORK INFORMATION CENTER TO THE AUGMENTATION RESEARCH CENTER (ARC) 5d

The NIC is presently a project intimately imbedded within ARC. ARC is an organization with multiple sponsorship which has as its goal the development of hardware and software computer tools, techniques, procedures, and training to aid man in his intellectual work. 5d1

The project has followed a research and development strategy of "bootstrapping", that is, of using the tools and techniques it has been developing in its own work, both as an aid to its

work and management and as a test "pilot plant" facility to try out ideas and techniques. 5d1a

As useful as this strategy is, there are limits to the type of feedback it can yield. The NIC is one of what we hope will be many projects set up to offer services to outside users. The goal is to provide a useful service and to obtain feedback on the needs of a wider class of outside users. We want to meet these needs with an integrated, modular system consisting of computer tools, people assistance, procedures, and training. We also hope to learn more about the problems of transferring augmentation services to a wide range of users. 5d2

The NIC consists of some personnel primarily concerned with its development and operation, but also draws heavily on the skills and work of most of the other members of ARC. As the NIC matures we are planning that it will grow into a well-defined semiautonomous cost center with more people specifically oriented toward its tasks. We want to clearly define the NIC's goals and needs. Where these overlap with those of other ARC activities, we wish to work closely on their realization and where they do not overlap to obtain the resources necessary to pursue them separately. 5d3

The long-run, future relationship between the NIC and ARC depends, we would guess, on the future operation of the ARPANET. The ARPANET may eventually be run by a commercial utility. If this happens the NIC could be transferred to that utility, become an independent enterprise, become a separate enterprise within SRI, or remain within ARC. The NIC is being developed to be more independent, so that its technology, procedures, and services can be moved if required. 5d3a

The Augmentation Research Center during its approximately 10 years of existence has been primarily a research and development organization providing service to itself rather than to outside clients. Therefore, along with development of NIC services has had to come a change in ARC's outlook, alterations in resource allocation, and changes in many of its practices, to enable it to offer a service and to maintain at the same time a vigorous R&D program. 5d4

OPERATIONS 5e

Computer Service Operations (CSO) 5e1

In the area of computer services, extensive measurement capabilities were added to the system to measure the efficiency of the TENEX operating system and NLS (#s,). A number of changes which appeared necessary as a result of these measurements were made and others are under study. 5e1a

Our hardware configuration contained a number of old, one-of-a-kind pieces of equipment brought over to the PDP-10 system from the previous XDS-940 system. These pieces of equipment have proven difficult to maintain and studies were launched on how to replace or upgrade this equipment. 5e1b

A new BBN network interface and a new DEC RP02 disc system were installed in the spring of 1972, replacing older unreliable equipment. 5e1b1

Hardware upgrading of our display system and its special core box has begun to provide temporary relief until a replacement system can be planned. 5e1b2

An additional 32K words of core has been added recently. 5e1b3

Studies leading to recommendations to add another channel, disc controller, and set of disc drives have been completed. These additions will provide more file storage capability and backup swapping capability. 5e1b4

The reliability improvements resulting from these measures and others under study should begin to be manifest in the summer of 1972. 5e1b5

Along with the above hardware improvements, improved practices and conventions have been evolved to handle new versions of software releases, both TENEX and NLS, and their checkout before being brought up for normal use. These conventions specify both frequency and time of day at which new systems can be brought up, and also specify documentation standards. 5e1c

One of the important aspects of CSO support has been implementation, integration and maintenance of those programs necessary for communication with the ARPANET and hosts connected to it. The basic Network Control Program and TELNET Protocols are obtained as part of TENEX support from BBN. When we had a non-standard hardware interface to the network and during early protocol development, considerable effort was

required in protocol implementation to create operating network programs. Less effort is now required, but this effort continues. We have also participated actively in working with the Network Working Group on protocol design and specification (proto qx,). 5e1d

People Services Operations (PSO) 5e2

During the past year ARC has developed several service functions that are now becoming operational for ARC users and NIC clientele. 5e2a

These functions (from activities such as RINS, NIC, Baseline Record, and Journal) and the forthcoming use of Deferred Execution (DEX) techniques have created new needs for people services support. 5e2b

As a result, we concentrated some of our effort on reorganizing these activities to allow more effective and efficient handling of routine and other tasks and to allow for easier expansion of the group size to meet needs of an increasing amount of throughput. The three aims were: 5e2c

To increase throughput to meet existing demands. 5e2c1

To become capable of expanding rapidly (in throughput quantity) to meet fluctuating service demands. 5e2c2

To work at minimizing costs while maximizing responsiveness to customer's needs and values. 5e2c3

This section describes in some detail the activities and tasks involved in setting up or running a PSO. We go into this detail because many people reading this report with a traditional computer service background may not appreciate the complexity of running an information service. Computer technology, while important, is not sufficient in and of itself to make possible such a service. Such a service is only possible with a balanced set of computer tools, people support services, and the methodology, procedures, and training which meld them together into an effective higher level system. 5e2d

Therefore in order to create such a balanced system we launched a new approach to ARC's "people services operations". (see -- 7834,1a) 5e2e

The main thrusts were: 5e2e1

organization
Physical Location and Configuration
procedure Establishment and Documentation
Transcription Activities
Terminals
personnel
Training

Organization 5e2f

A group with skills in handling paperWork and messages, and in using TNLS and DEX, was explicitly identified as PSO, and a group of advisors with skills in administration, documentation, and training was assigned to assist in getting PSO into formal operation. 5e2f1

Physical Location and Configuraion 5e2g

Office and workroom areas were expanded and relocated, to give the growing support operations more efficient location and arrangement. New tables, shelves, cabinets, and files were acquired and their arrangement worked out. 5e2g1

FIGURE 21. People Service Organization workroom.

5e2g1a

Procedure Establishment and Documentation 5e2h

procedures were devised and documented for: 5e2h1

Use of TNLS (see -- 7470,) and DEX (see -- 9934,), 5e2h1a

The handling of transcription and other service requests. 5e2h1b

All related NIC activities -- clerical and secretarial. 5e2h1c

Transcription Activities 5e2i

Types of work to be handled: 5e2i1

Handwritten drafts
Tape recordings
Dictation notes
Offline documents
Online documents to be edited

Techniques for transcribing material into online files were developed: 5e2i2

Deferred Execution (DEX) covered at greater length see --,4d1) 5e2i2a

This process makes use of terminal and magnetic tape recording equipment for initial input of data with actual entry into computer files deferred until periods of low system use (thereby resulting in less expensive use of the system for the processing of this work.) This system has been used to place online many documents of importance to the ARPANET community originally prepared offline. 5e2i2a1

Where and how long to store entered tapes for backup, the conventions for hierarchical statement entry, and when the transcriber should try to put hierarchical structure into documents are still under development. 5e2i2a2

TNLS 5e2i2b

TNLS is used largely for routine editing of online documents, and for entering high-priority items during off-peak load hours. 5e2i2b1

DNLS 5e2i2c

Display NLS is used for difficult editing of online documents and for some highly formatted documents. 5e2i2c1

Receiving processes 5e2i3

We set up a central receiving station. 5e2i3a

One person, with an alternate, handles users' questions regarding job status, time and cost estimates, etc. 5e2i3b

Priority determination process 5e2i4

A requestor specifies his preference for priority: 5e2i4a

Immediate service (1-4 hours)
Normal service (4-12 hours)
Deferred service (a week or two)

Temporary storage of unassigned work 5e2i5

A log system using appropriate work request forms has been set up. 5e2i5a

We have a central storage place, organized for control of work by priority. 5e2i5b

Assignment process for transcription work 5e2i6

A work scheduler assigns incoming work to group members, balancing priority request with members' capabilities and workload. 5e2i6a

Later, priorities may be established by a bidding scheme. 5e2i6b

We plan to enlarge this effort to allow assignment to an outside pool of workers trained in DEX, both SRI people and contract manpower. 5e2i6c

Output processes 5e2i7

We have developed conventions for naming of temporary input files (special and separate for the catalog process) with provision for special instructions from the author. 5e2i7a

We have developed procedures for delivery of completed work to the requestor. 5e2i7b

Terminals 5e2j

We have made a thorough study of available teletype terminals and magnetic tape devices, and after experimental use of several, have leased nine TI terminals and six Termicettes, for use with DEX. 5e2j1

Personnel 5e2k

We have added several new staff members with contributions to make to NIC. Two writers who can also teach were active in PSO development. Three new staff members were added to the document preparation, transcription and distribution efforts. 5e2k1

Training 5e21

Classes in TNLS and DEX were held for ARC and network people. Manuals were prepared. A more detailed discussion of training is given later, see--,5g10). 5e211

A detailed list of the types of tasks this PSO group and associated information handling people perform to support the NIC is given below because it is important for people to understand the range of activities that are required even with automated aids to support a service such as the NIC. 5e2m

PLANNING AND SCHEDULING 5e2m1

- Goal setting 5e2m1a
- Service design 5e2m1b
  - Site Station aid planning
  - Functional document design
  - NIC Collection design
  - Station collection plan
  - Reference service design
- Catalog design 5e2m1c
- Procedure establishment 5e2m1d
  - Discussion
  - Procedure writing
  - Experimentation
- NIC facility design 5e2m1e
- Work flow scheduling 5e2m1f
- NIC time and cost studies 5e2m1g

GENERAL SUPPORT 5e2m2

- Dictation
- Phone
- Orders and financial records
- Timecards
- Visitor arrangements
- NIC travel arrangements
- NIC facility upkeep

STATION PHONE ACTIVITY 5e2m3

- Station phone answer
- NIC outgoing calls

MAIL SINGLE NIC PIECES 5e2m4

- Incoming mail processing
- Single mailings

ACQUISITION OF NETWORK INFORMATION 5e2m5

- Network resources
- Network personnel
- Network publication references

CHOICE, ANALYSIS, OF INFORMATION 5e2m6

- Analysis for bulletins
- Analysis for functional documents
- Selection of publications
- Abstracting

ACQUISITION OF PUBLICATIONS 5e2m7

- Checking holdings
- Order form preparation
- Receipt, record changing

OFFLINE CATALOGING WORK 5e2m8

- Coding
- Checking of coding, revision
- Proofing and revision
- Recoding of old material
- Catalog offline records
- Old catalog offline work

FILE-BUILDING ONLINE 5e2m9

- Input of new citations
- Input of old citations
- Editing of new citations
- Editing of old citations
- Bulletin creation
- Bulletin editing
- Catalog creation
- Catalog editing
- Catalog file manipulation
- Functional documents input

Mailing list input
Letter online input
Other online text input
Other text input, DEX
Identfile maintenance

PHYSICAL PROCESSING 5e2m10

Readying of Journal printout
Readying of other work
Collating
Stamping, Punching
Xeroxing of documents
Line printer output
Outside repro contact

DISTRIBUTION 5e2m11

Mailing list maintenance
Labels, envelope preparation
Pickup and delivery

STORAGE AND MAINTENANCE 5e2m12

NIC Master collection
Extra copies
Supplies

VISUAL AIDS 5e2m13

Chartmaking

TRAINING 5e2m14

Instruction
Development of training aids

REFERENCE WORK 5e2m15

Locating citations for Net
Locating documents for Net
Literature search

Let us now look at each of the services provided and see what has been involved in making them available beyond the changes

described above, why they were made available, and some future plans. 5e3

There are two major areas of changes to ARC caused by providing NIC services that deserve mentioning: planning and providing more reliable and efficient computer services, and planning and providing more varied and extensive clerical and other services provided by and for people. 5e4

ONLINE SERVICES 5f

ACCESS TO NLS 5f1

The ARC ONLine System (NLS) is an evolving system which we view as an integrated set of tools for doing general intellectual work (,4d1,). To this end NLS has, at this time, powerful document creation, editing, production, and studying capabilities, dialog support functions for online communication both simultaneous and distributed in time, bibliographic catalog-making capabilities, programming aids and facilities, some basic information retrieval abilities, and some, as yet, rudimentary management and other planning aids. 5f1a

A subset of these capabilities, felt to be of prime value to initial NIC use centered around document creation, editing, production, and studying as well as dialog support, has been thoroughly documented for NIC clientele. 5f1b

We knew that most systems on the network supported typewriter terminals rather than displays so that during the conversion from the XDS-940 to the PDP-10, a typewriter version of the system was designed and implemented --,4d1a3a). 5f1c

In thinking about the problems which could exist in supporting all the varieties of typewriter terminals on the network, we (to keep ARC's thoughts clearly separate from the net's) felt that it would be better to have most of these differences handled by a standard network protocol. Therefore, we worked actively with the Network Working Group (NWG) in establishing a network virtual terminal protocol (TELNET) see--,6b1). This protocol has succeeded in allowing access to TNLS from different systems and terminals. 5f1c1

NLS, as it has historically developed, is oriented in its command language design for expert users. 5f1d

This orientation toward highly trained, experienced users is not completely suitable for the clientele of the NIC, comprised of some who use the system often enough to become experts and others who want to use the system infrequently. Therefore, thought has recently gone into studying what changes are needed in the NLS command language syntax to provide a range of modes from novice to expert, and what additional help and tutorial capabilities need to be built into the system. These changes will be implemented in the coming months. 5f1d1

At the present time anyone with access to a typewriter terminal connected to the network and with an entry in our identification file ,see--,4a4) (entries can be made in this file directly by network users) can have access to NLS. 5f1e

We generally find between 1 and 3 users from the network using TNLS during prime hours. The highest number observed has been 7 simultaneous network users. The number of logins a day from the network has been averaging around 40 - 50, with a variation between 30 and over 100. We expect the number to increase significantly both as the network grows and as our hardware reliability improves, as discussed earlier. 5f1e1

A system for allowing access to the display version of NLS (DNLS) from the network using IMLAC display terminals equipped with a keyset and mouse has been developed jointly with the Xerox Palo Alto Research Center and tested with users from UCLA-NMC and BBN-TENEX. 5f1f

We expect to continue experimental use of DNLS over the network and eventually to offer DNLS as a regular service. We are currently studying how to provide DNLS service from low-cost alpha-numeric displays equipped with keyset and mouse. 5f1f1

Documents are presently created by a user at a keyboard device connected to TNLS via the network. We are working to allow entry of documents into NLS which were initially prepared in other host computers. At least one site, MIT-DMCG, has been entering documents in NLS by preparing locally a file of NLS

commands and document text and transmitting it into NLS as a simulated teletype. The schemes under study will simplify this process. 5f1g

AIDS TO COLLABORATION 5f2

We envision a wide variety of collaboration aids to help geographically distributed people work closely together. One such system being developed and offered as a NIC service is the Dialog Support System (DSS). The first steps in the creation of a DSS have been taken in the implementation of Journal, Number, and Identification systems. 5f2a

As discussed in more detail above, the Journal is a system for capturing recorded dialog items (in the form of documents and messages) and for distributing these items online, offline, and through the network to the appropriate recipients. 5f2b

When an item is submitted to the Journal, a unique number is associated with it, either obtained at the time of submission or previously from the Number system. This unique number is used for cataloging purposes and as the name of the item for later reference and retrieval. 5f2c

Once submitted, the items become read-only; statements in a Journal item can be uniquely and precisely referenced in future documents with assurance that the reference will remain meaningful. 5f2d

At the time of submission, or any time later, documents can be distributed to one or more individuals, either singly or as members of groups by indicating to the system a list of unique identifications called IDENTS. 5f2e

New identifications can be created at the time of submission or at other times by use of the Identification system. 5f2e1

The IDENTS are usually a person's or group's initials. The IDENTs are automatically assigned by the Identification System when a person's or group's name is entered into an identification file by use of the system. 5f2f

When one is sending an item to a group, one need only use the group's IDENT and the system will deliver to the the

membership of the group. One can also indicate distribution to only the coordinator of the group. 5f2f1

If one does not remember a person's or group's IDENT at the time of submission, a query capability allows it to be retrieved. 5f2g

The Identification System has provisions for collecting other relevant information such as a phone number, network site affiliation, and preferred method of document delivery (online as a citation in the receiver's Initial File, offline by hardcopy through the mail, or both). 5f2h

The information in the identification file is used by the Journal System during document submission and delivery. The information in the identification file is also used to automatically prepare directories of individuals and dialog groups as described later. 5f2i

SAMPLE MESSAGE SENDING SESSION 5f2j

The following is a demonstration of how a message is submitted to the Journal by a Network user (including login, NLS access, and logout procedure). Material in square brackets is fed back by the system. Material in parentheses is commentary. The symbols *, @, &, && are system heralds and are not shown in brackets. 5f2k

@Login SP DOE SP DDD SP 1 CR (A user named DOE logs in to the system - nis ID is DDD)

/JOB 11 ON TTY14 3-AUG-71 17:11/

@nls CR (The user accesses the NLS system)

/ID:/ DDD CR

/device:/ N/et-tty/ ("N" signifies that the user DDD is connected through the Network with local echoing)

*e/xecute/ j/ournal/ (access Journal system)

/submit/ m/essage/

SRI-ARC 8 JUNE 1972 13041
Network Information Center
Development and Operations
Online Services

THIS IS A SAMPLE MESSAGE CA (Contents of message; note "CA" means Command Accept)

/number/ CA /3333/ (typing CA after request for number causes system to assign DDD a unique catalog number /3333/ for the message)

&&i/nterrogate/ CA (causes the system to prompt the user for the correct information required to send the item through the Journal)

&&/title:/ A SAMPLE JOURNAL SESSION CA

&&/distribution:/ XXX (for your action) YYY (for your information) CA (XXX and YYY identify other persons known to the system; the text in parentheses are comments directed to them.)

&&/status/ CA

.

. (the system prints back all information entered by the user)

.

&&/go?/ n/o/ (the user does not want to "go" now but wants to add more information not requested by the Interrogate procedure)

&&k/eywords:/ test sample CA (keywords provide input to an index to all messages)

&&c/omments:/ isn't this fun! CA

&&g/o?/ y/es/ (tells the system to begin Journal process. Note that as the author command is not used, DDD is assumed).

/JOURNAL SYSTEM IN PROGRESS/ (the system is processing the new Journal entry)

/Journal,JRN1,J333:gw/ (the system has created a link to the new Journal entry)

e[xecute] q[uit] CA (the user has been returned to the NLS command level as signified by the system typing the herald character "" the user then types the Execute Quit command to return to the EXEC)

@logout CR (the user logs out)

EXAMPLES OF ONLINE JOURNAL DELIVERY 5f21

When Journal items are delivered to a person, they are delivered (as citations for documents, or the actual text for messages) in a file called his "initial" file as it has as a name the person's IDENT or initials. The citation contains the IDENT(s) of the author(s), the date and time catalog number; the title on a second line; and an NLS "link" or the message on the third or additional lines (see discussion below on links).

<WATSON>RWW.NLS;372
*Print Branch .1!w!
(Journal) Journal Documents (most recent first)

DCE 31-MAY-72 10:01 10614
Comment on user-feature change coordination, and (10587,)
Location: (JJOURNAL, 10614, 1:w)

LPD 30-MAY-72 10:39 10591

Message: CAN YOU SEND ME A COPY OF THE LATEST LIST OF HOSTS?
RAY TOMLINSON SAYS THE NCC ISSUES AN RFC ON THIS SUBJECT PERIODICALLY.

Delivery of hardcopy of computer-processed documents is not yet as smooth as we desire and takes longer than desired because of the chain of events that must presently take place in this process. The present chain of events is: 5f2m

1) creation of an image of the documents for each receiver on magnetic tape. 5f2m1

2) Transfer of the tape to SRI's computer center for batch printing. 5f2m2

3) Document printing. 5f2m3

4) Bursting. 5f2m4

5) Stapling. 5f2m5

6) Checking for correct content and addresses. 5f2m6

7) Mailing. 5f2m7

The documents are printed with a cover mailing address sheet. We are presently working to improve the reliability and scheduling of the above chain of events. 5f2n

We expect in the next few months to be automatically delivering documents through the network for printing at the destination sites or for delivery to online files. 5f2o

This will require ratification of at least an experimental file transfer protocol by the Network Working Group and implementation of this protocol. Such a protocol is presently under development. 5f2o1

We expect, as mentioned earlier, to allow documents prepared on local host text editors to be entered into the NIC Journal and be automatically cataloged and delivered by this system. 5f2p

To uniquely identify hosts and NIC stations associated with the network we worked with the NWG to set up a standard identifying syntax and asked each host and NIC station to name themselves according to the established rules see--,6c1)? 5f2q

This was a small but important step to help establish uniform communication rules useful to different processes of the ARPANET experiment such as various NIC services. These host IDENTs are dialog groups. Thus to send an item to everyone at ARC one uses the IDENT "SRI-ARC" in the distribution list at the time of Journal submission. 5f2q1

At the time of submission of a Journal item a catalog entry is created with all the relevant information such as number, author(s), title, date-time, documents obsoleted or updated by this document, person who actually did the submission, keywords if any, distribution, affiliation of author(s), and subcollections. 5f2r

Each dialog group or affiliation is a subcollection name as well, and all items sent to that group are automatically part of that subcollection. 5f2r1

All the above information can be explicitly entered at submission time. It is from this catalog information that indices and listings are made periodically as described later. 5f2s

Users can find dialog items of interest by use of the catalog listings and indices. 5f2t

Thus, by use of the Journal and the catalog of Journal items, people can find and participate in dialog distributed over time and being carried on by people geographically separated from one another. See the discussion in the next section for a sample query of the catalog. 5f2u

One of the features of NLS is a link mechanism. An NLS link is a syntactic entity which references a statement in the current document, or in any other document. The link can also control the initial view of the referenced item. 5f2v

NLS has mechanisms which allow one to "point" at a link and have the system fetch and display the item referenced. 5f2v1

Thus, using links, networks of related documents and dialog items can be created. 5f2v2

The system saves the last several documents (and positions within them) examined, and thus one can move ahead to an item and then, when the appropriate command is given, return to previous positions automatically. 5f2v3

Links are essentially forward references. At the present time one cannot automatically access those documents pointing to a given document (i.e., if one is in a document he cannot now ask what documents reference it). 5f2v4

Plans to implement this "backlink" capability are being made. This facility will add considerable power and a citation chaining and indexing capability. 5f2v4a

We also plan to implement a comment capability which will allow persons studying a document to easily comment on

dialog items. Others will be able to selectively view these comments. 5f2v5

It is expected that in time NLS will be run on several PDP-10's in the ARPANET. Each of these systems will have a journal. One of the coming research and development problems will be to create a network of cooperating journals which allow documents to be distributed throughout the network, but be entered or retrieved from any system. 5f2v6

Once this problem is solved, further generalization to other non-PDP-10 hosts can be made. 5f2v6a

ONLINE ACCESS TO STANDARD NIC DOCUMENTS 5f3

Access to NIC documents is handled with the general NLS mechanisms presently existing. These mechanisms combine to give a powerful query and browsing capability to those users trained in NLS usage. These mechanisms, however, are not satisfactory to users who are unfamiliar with NLS usage. Since we can always anticipate occasional users and users new to the network, it is planned to implement a novice-mode query capability for these users. 5f3a

Many interactive, online query systems exist which allow one to specify a query by certain keywords or phrases, and logical combinations of these. 5f3b

One, the MIT TIP system, also allows a citation chaining and query capability. There is a great deal of interest in the information sciences field in designing interactive retrieval systems with the proper user interface. Few of these systems also allow full document retrieval as well. In spite of the many prototype and experimental retrieval systems in existence, one gets the distinct impression in talking to people who have used these systems and from the literature that there is much development and much to be learned in this area. 5f3b1

We feel that capabilities such as keyword retrieval and citation chaining are important and useful, but that other capabilities such as catalog browsing, document browsing, and studying document editing and creation are necessary parts of a complete document handling and recorded dialog retrieval process. 5f3c

NLS has the potential to serve as a basis for such a fully integrated system. Because of the desire to fully understand and provide the implementation foundations of NLS to tap this potential and the desire to build on the large amount of work in the retrieval field, we have taken a "go slow" approach to the query problem and have built a simple but powerful accessing capability using presently available NLS mechanisms. 5f3d

The mechanisms used are the NLS link mechanism described earlier, search by statement name or content, and use of view specifications. Before presenting a sample catalog query session we outline some thoughts on an initial query capability. 5f3d1

The standard NIC documents available online are: 5f3e

1) TNLS, Journal, Identification, and Number System User Documentation 5f3e1

2) Some workbooks for aiding people in learning TNLS 5f3e2

3) The ARPA Network Resource Notebook, describing facilities available at each site which are offered to the network community 5f3e3

4) Catalog of Listings and Indices to the NIC collection of network dialog and network related documents 5f3e4

5) Current Directory of Network Participants 5f3e5

6) Soon the Current Network Protocols document will also be available online 5f3e6

Users may access and query this collection of information using standard NLS capabilities by use of a master document contained in a file called <NIC> Locator, a copy of which is contained below. This Locator document contains pointers to the various sections of the documents listed above in the form of NLS links. Locator is a form of inverted file. Once having arrived in the desired document, by taking the appropriate link, one can use NLS printing and view specifications to browse, or one can search for a desired point by content or appropriate keyword. A copy of the Locator User Guide is included as appendix A

. 5f3f

SAMPLE SESSION using <nic>locator 5f3g

Retrieval as mentioned earlier is by use of preset NLS mechanisms. System printout is shown in light face, comments are in Times Roman italics enclosed in parenthesis. <V2=8p,5,Light+Slanted>

Load File <nic>locator!

*Print Branch .2!xbm! (Printout of an appropriate view of the NIC documents)

2 NIC DOCUMENTS

2A NIC TNLS USER GUIDE pages=112
2B NIC JOURNAL USER GUIDE pages=78
2C NIC TNLS EXERCISE FILES pages=23
2D CURRENT CATALOG OF THE NIC COLLECTION pages=404
2E CURRENT DIRECTORY OF ARPA NETWORK PARTICIPANTS pages=133
2F ARPA NETWORK RESOURCES NOTEBOOK Pages=62
2G CURRENT NETWORK PROTOCOLS (not yet implemented online)
2H FOLKLORE...day to day information on NLS pages=11

*Print Branch .2d!eb! (Selection of the catalog and printout with more detail)

2D CURRENT CATALOG OF THE NIC COLLECTION pages=404

2D1 INDEX BY AUTHORS pages=69
2D2 INDEX BY TITLE WORD pages=333
2D3 RFC LIST BY RFC NUMBER pages=17
2D4 NIC INDEX BY NIC NUMBER pages=121

selection of the author index.

* .2d1 ↑! (Search for documents by Watson)

**Print Group .watson! .watson t!

1192 (Watson) NWG/RFC 289: What We Hope is an Official

SRI-ARC 8 JUNE 1972 13041
Network Information Center
Development and Operations
Online Services

```
List of Host Names                    21 Dec 71    8295
Watson

1193 (Watson)* Reply to JBL on Output Device Teletype
               20 Dec 71     8289     Watson

1194 (Watson)  Summary of 1971 Activities
                  9 Dec 71     8158     Watson

1195 (Watson) NWG/RFC 280: A Draft Set of Host Names
               17 Nov 71     8060     Watson

1196 (Watson) NWG/RFC 278: Revision of the Mail Box
Protocol /See Number Listing/

↑o      (STOP PRINTING)

* &!     (Return to Locator)

<NIC>LOCATOR.NLS;1

*Print Branch .2f!!      (Printout contents of Resource
                          Notebook)

   2F ARPA NETWORK RESOURCES NOTEBOOK  Pages=62

      2F1 INDEX pages=23
      2F2 BBN-TENEX  pages=10
      2F3 CASE pages=5
      2F4 CARNEGIE  pages=5
      2F5 HARVARD-1  pages=3
      2F6 HARVARD-10  pages=7
      2F7 ILLINOIS pages=5
      2F8 INTRO  pages=8
      2F9 LL-67  pages=3
      2F10 LL-TX-2  pages=15
      2F11 MIT-AI  pages=3
      2F12 MIT-DMCG  pages=7
      2F13 MIT-MULTICS  pages=15
      2F14 RAND  pa

↑o      (STOP PRINTING)

   *Print Branch .2f12 ↑!!      (Selection of the MIT-DMCG
                                 entry)
```

SRI-ARC 8 JUNE 1972 13041
Network Information Center
Development and Operations
Online Services

<LISTER>MIT1.NLS;7, 3-MAY-72 16:24 PL ;.HJOURNAL="NIC 9894

(Arrive with appropriate viewspecs to see table of contents)

1 I. Personnel
2 II. Installation Type
3 III. Equipment
4 IV. Physical Resources
6 VI. Interests and Capabilities
7 VII. Login
8 VIII. Computer Operator
9 IX. Miscellaneous
10. X. Programs

<NIC> MIT-DMCG.NLS;2 (Feedback from system as to destination)

*Print Branch .7!w! (Selection of the Login information)

7 VII. Login

7A A. When implemented, the logger would be in accordance with standard Initial Connection Protocol utilizing "socket 1" for connection. The final full-duplex connection would involve sockets US+2 and US+3.

7B B. As soon as the full-duplex connection is established, the system would send to the user the following ASCII characters (7-bit ASCII, 8th bit zero):

7B1 MONIT.MN CR-LF
;
where MN is current version of MONIT and CR-LF are ASCII characters carriage return and line feed. The user should then transmit the following ASCII character string LOGIN <name> CR
where <name> consists of a maximum of six ASCII letters or numerals. (The system at command level does not distinguish between upper and lower case as it maps them into 6-bit characters.) We ask

that the name be ASCII characters in the following order: Host site number followed by user's initials. Upon receiving login the system will respond with the prompt character:
;

Now the user is logged in and can use the system. To logout the user may simply send the command
LOGOUT CR
The system will then respond with an appropriate message. Following the receipt of this message, the user should ask his NCP to close the full-duplex connection.

Below under Offline Services we describe in more detail the concept of a functional document and the processes involved in creation of the NIC standard documents. 5f3h

OFFLINE SERVICES 5g

Introduction 5g1

At each network site and at some sites without computers connected to the network there is a NIC Station consisting of a Station Agent, a Technical Liaison, and a NIC Collection. There are presently 56 NIC stations of which 4 are outside the USA. 5g1a

The Station Agent's job is to maintain the NIC Collection for a site and be familiar with various NIC procedures to assist people at the site in use of the NIC. The Technical Liaison's role is to be familiar technically with his site and usually also to participate in network development and use. 5g1a1

The NIC maintains a master collection at SRI where items felt to be of use to each site are reproduced and distributed to the site's local collection. Liaisons also receive copies of some network dialog of interest to them and also receive updates to Functional Documents. 5g1b

This concept of a master collection and an associated set

of satellite collections is an important part of the NIC operation. This satellite collection operation needs more work and design than we have yet been able to give. Four areas needing work are: 5g1b1

We need to provide more and better training to Station Agents on how to handle the satellite collections. 5g1b1a

We need to evolve our cataloging and catalog production tools to the point where Stations can maintain their own subcollections and shelf lists. 5g1b1b

We need to provide selective dissemination of documents to Stations based on interest profiles of users at that site. 5g1b1c

We need to investigate production and use of microfilm technology. 5g1b1d

Functional Documents and Their Revision 5g2

INTRODUCTION 5g2a

several documents generated in Network activities are subject to occasional revision and updating. The CURRENT CATALOG OF THE NIC COLLECTION, the DIRECTORY OF NETWORK PARTICIPANTS, and the NETWORK RESOURCE NOTEBOOK are examples. These and external documents such as the BBN manuals are referred to by NIC as "functional documents". 5g2a1

More generally, a functional document is a document whose title and function remain constant, but whose contents can change. A functional document contains a single or several documents which can be added to, deleted, or replaced entirely or selectively. Thus the functional document, which has a NIC number, can be referenced in other documents with some assurance that it will be in existence, even though the subdocuments with their distinctive NIC numbers may be in flux. In the Catalog the number of a functional document in which a specific document may be contained is listed, and the current contents of each functional document is indicated. 5g2a1a

To illustrate, the NETWORK RESOURCE NOTEBOOK may

always be referred to as NIC 6740. To allow the descriptions of individual sites to be updated separately, each section is uniquely numbered, and is then renumbered each time it is reissued. 5g2a1a1

Another functional document, CURRENT NETWORK PROTOCOLS, NIC 7104, was established to bring together all currently active documents on ARPA Network Protocols. Its contents may include documents also issued separately. 5g2a1a2

Each functional document has a Contents Page which shows the names and numbers of the content documents as of the date it carries. It has also a Status of Contents page which gives information on documents superseded, and the dates of revision of all documents and of any individual pages revised. Further information for use in tracing the history of the contents is contained in the series of transmittal letters sent with partial contents, as discussed below. 5g2a1b

In preparing a document which is expected to be revised, Network participants are urged to use a looseleaf format. 5g2a1c

The Network Information Center intends to support the distribution and recording of contents of functional documents. Procedures have been established, as described below, for fitting the changes to such documents into the NIC system, and for reproducing and distributing them to individuals or stations with instructions for their integration into the existing documents. 5g2a2

PROCEDURES FOR REVISION MATERIAL 5g2b

original manuals and other functional document materials are reproduced and distributed by NIC just as other Network publications. For all documents obtained through NIC, NIC attempts to receive and make distribution of updates. 5g2b1

Inclusion of an additional document in a functional document: 5g2b2

If the added document has already been distributed separately, the holder of the functional document may

sometimes be asked to insert the old copy in the functional document. Usually a new document or copy is supplied. 5g2b2a

substantial revision of a bound document, or of more than a few pages of a loose leaf document: 5g2b3

A new document is published, with a new NIC number. It bears a notation under the number on the title page and/or cover, e.g., 5g2b3a

NIC 5772
supersedes NIC 5621 5g2b3a1

few pages inserted or REVISED IN A LOOSELEAF OR CORNER-stapled document: 5g2b4

Each new or revised page bears the original document number, with a date of revision, e.g., 5g2b4a

NIC 5742
3-OCT-71 5g2b4a1

Inserted pages are numbered to fit into the existing document, e.g., pages 5.1, 5.2, 5.3, may be inserted between pages 5 and 6. 5g2b4b

Deleted pages are replaced by a single page indicating the deletion, e.g., 5g2b4c

Pages 7-12 deleted, 25-MAR-71 5g2b4c1

Revisions are made only by substitution, addition or deletion of a full page or more. NIC does not revise its own publications by lists or errata, and strongly recommends against their use by others in the Network. However, when NIC receives such lists of errata, it reproduces and distributes them with suggestions to Station Agents for recording and inserting them. 5g2b5

DISTRIBUTION AND TRANSMITTAL PROCEDURES 5g2c

The transmittal letter accompanying a set of revision material and the revision material itself constitutes a

separate document, a copy of which is filed at NIC, where a new copy can be provided at any time. 5g2c1

The transmittal letter indicates the appropriate information: document number of the revision material, date, document number of the publication being updated, its date, and, when practicable, information on the changes which the revision describes or implements. 5g2c2

Each functional document has a section at the end for filing the transmittal letters accompanying the contents. These transmittal letters are numbered sequentially as well as with NIC numbers, so that the sequence of changes can be established. 5g2c3

PLANS 5g2d

At the present time the tables of contents of functional documents must be maintained by hand, although plans exist to develop automatic aids associated with the Journal for production and maintenance of functional documents. 5g2d1

Building a Network Reference Bibliographic and Dialog Data Base 5g3

COMPUTER-PRODUCED CATALOGS AND INDICES 5g3a

Introduction 5g3a1

We have directed effort toward the development of a Catalog Support System (CSS), needed initially to support clerical processes for maintaining current online catalogs of the Master Collection and several subcollections and for producing various indices (hardcopy and online) to these collections. Subsequently, support will be needed for augmenting various online user-level information-handling processes. 5g3a1a

The CSS is concerned with the following principal processes: 5g3a1b

Input, editing, proofing, and verification of catalog entries. 5g3a1b1

Updating of the Master Catalog and subcollection catalogs. 5g3a1b2

Production of incremental and cumulative, hardcopy and online indices to various collections. 5g3a1b3

Overall Design Goals and Elements 5g3a2

The basic goals relevant to providing aids to these processes are: 5g3a2a

Maintaining integrity of the master catalog files with maximum protection from both human and mechanical errors. 5g3a2a1

Making possible a smooth flow of input from ARC clerks with good facilities for proofing and correcting all clerical input. 5g3a2a2

Removing as much load as possible from the computer system during prime use times through the use of Deferred Execution techniques. 5g3a2a3

Providing an NLS subsystem which integrates well into the rest of the NLS system and can be used by other file processes as well as those required for catalog production. 5g3a2a4

The initially implemented element of the Catalog Support System is the Catalog Production Processor (CPP). 5g3a2b

The CPP is the basic output port of the CSS and is designed to allow the production of online and offline, incremental and cumulative, indices and listings of various kinds, using the Master Catalog as the ultimate data base. 5g3a2b1

One objective in the design of the CPP, in fact of most of the CSS, is to avoid adding new basic capabilities to our augmentation system, but rather to bring together existing ones in such a way as to reduce our commitment of resources to clerical tasks. 5g3a2b2

The basic design goals which the CPP must meet are the following: 5g3a2b3

It should permit flexible specification of the types and frequencies of production of the various catalog indices and listings needed by DSS, NIC, DPCS, etc. 5g3a2b3a

It should function as automatically as possible and with a minimum consumption of ARC personnel and equipment resources. 5g3a2b3b

The CPP implementation has now progressed to the testing stage and is expected to be used experimentally in the production of the ARC Journal and NIC Catalogs in the next few weeks. 5g3a2b4

Use with the entire ARC Master Collection as an aid to completely integrating the various subcollection citations will follow. The CPP will then be available for ARC use on any desired subcollection catalog-production work, either Journal, NIC, or special subsets. 5g3a2b4a

SELECTION OF ADDITIONS TO THE DATA BASE 5g3b

The ARC Master Catalog is a group of files containing the catalog-entry statement for all informational items that we hold for purposes of control, retrieval, and access. The NIC collection is a subcollection of the ARC Master collection. 5g3b1

Active experimentation in the collection of information items and interaction and connection with other existing data bases and information services is still in the future plans of ARC and NIC. However, during the past year ARC took the opportunity to input the contents of some data bases gathered elsewhere, and to output the contents in new formats. 5g3b2

Data bases thus added include: 5g3b3

A bibliography prepared for use of the attendees at the January 1971 AFIPS Workshop on the User Interface. The bibliography and indices processed by ARC programs were published in the volume of Proceedings of the Workshop. (see -- 9474,) 5g3b3a

An extensive bibliography on networking prepared by Peggy Karp of MITRE. Each reference cited was obtained in full-size copy and was coded and entered in the Master Catalog. (see -- 6025,) 5g3b3b

special "subcollection catalogs", such as for the NAS Information Sciences Panel, for the AFIPS Workshop, for the ARC Journal or for the Network Information Center, are created by (automatically) collecting a copy of every entry statement in the Master Catalog having a descriptor code of NAS, AFI, JOU or NIC respectively in its "z2 field." 5g3b4

DESIGN OF DATA ELEMENTS 5g3c

The usefulness of a data base of citations to information items depends on the elements of data selected to describe the items. The selection criteria and their implementation become even more important when the items of information include forms of information other than published books, articles, and reports,, e.g., films, slides, letters, photos, ads, meeting agenda, maps. 5g3c1

A continuing effort has been the refinement of a set of data elements. The requirements are: 5g3c2

Data elements should be adequate to describe all species of information items which are anticipated to be added to the collection. 5g3c2a

Data elements should be adaptable to economical use by programs developed for gathering and formatting the citations into catalogs and listings and for online retrieval. 5g3c2b

The present list of data elements and guidelines defining their application is appended. (see -- 9868,). Future development will study the appropriateness of using standard data elements being designed by national and international committees considering bibliographic data. 5g3c3

ENTRY OF ITEMS INTO THE DATA BASE 5g3d

As noted, items of information relevant to NIC appear in many forms. Reference to these items is simplified by assigning a master catalog number, a serial number, to

each. To record the items to which the catalog number refers, a description of the item using the data elements noted above is coded by ARC and entered as a "statement" in an NLS file. 5g3d1

procedures necessary to ensure a consistent, clean data base are vital and difficult to hammer out. Much effort has gone into this area over the past year. 5g3d2

An example of a catalog-entry statement with typical coded data elements: 5g3d3

(M4623) *a1 Howard Frank #2 org *b2 Network Analysis Corporation #4 Beechwood, Old Tappan Road #5 Glen Cove. New York 11542 *c1 First Semiannual Technical Report for the Project Analysis and Optimization of Store-and-Forward Computer Networks #6 62p. *d1 15 June 1970 *d3 15 October 1960 - 15 June 1970 *f1 r *f2 o *s1 ARPA #6 DAHC15-70-C-0120 #7 0D30 #8 1523 *w1 6-30-71 *y1 Discussed analysis and optimization of the ARPA Computer Network, general design philosophy. Relationships between traffic level, link capacities, and cost as a function of number of nodes in the network have been investigated. Extensive studies made for 12, 16, 18, 20 node networks, where each node was a potential site. *y2 network analysis; computer networks; store-and-forward networks; topological optimization; *z2 NIC *z3 new * 5g3d3a

DESIGN OF CATALOG FORMATS 5g3e

A set of special programs has been written at ARC to collect, sort, analyze, and reformat the entry statements to produce catalogs and indices such as those in the current Catalog of the NIC Collection, (see -- 5145,) and those used in NAS Panel and AFIPS Workshop meetings. 5g3e1

These programs, described below, are the result of much thinking and experimentation to produce catalogs and indices of maximum usefulness, given the present printing constraints. 5g3e2

Examples of the listings and indices now produced are: 5g3e3

SRI-ARC 6 JUNE 1972 13041
Network Information Center
Development and Operations
Offline Services

Catalog listing by number: 5g3e3a

The Validity of Basing Computer Selections on Benchmark Results. 6557

Edward O. Joslin and John J. Aiken (U.S. Air Force Systems Command, Electronic Data Processing Equipment Office, Hanscom Field, Bedford, Massachusetts).

Computers and Automation, Vol. 15, No. 1, p.22-23. January 1966.

Author Index:

Title	Date	Number	Author
The Validity of Basing Computer Selections on Benchmark Results	Jan 66	6557	Aiken

Titleword Index:

Title	Date	Number	Titleword
The Validity of Basing Computer Selections on Benchmark Results	Jan 66	6557	Validity

Number Index:

Author	Title	Date	Number
Joslin	The Validity of Basing Computer Selections on Benchmark Results	Jan 66	6557

PROBLEMS ENCOUNTERED IN BUILDING A REFERENCE DATA BASE AND NIC CATALOG 5g3f

The steps involved in building an online data base and machine produced catalog have shown us that successful operation of such a system requires well-trained staff, reliable computer system operation, carefully worked out and documented procedures, careful proofing and just plain luck. Experience has shown that a full blown augmentation subsystem such as our bibliographic reference system contains a full mixture of computer

tools, people, procedures, and training and that integration and development of such a system is a non-trivial process. 5g3f1

We have found even with our small collection of less than 2000 items that the period between issues of new catalogs tends to be about once a quarter. Our goal when the new Catalog Production Processor is fully integrated into the present system is to produce a catalog every 4-6 weeks, with weekly announcement bulletins of new additions to the collection. 5g3f2

The problems of the printed Catalog are not unique to this document; they occur also in the preparation of the Directories and will occur in some form in other functional documents, but the diversity of the data elements and the complexity of the formatting are greater with the Catalog than with other documents. 5g3f2a

NIC has had the experience, common to other information centers, that bibliographic processing entails more effort and more sources of delay and difficulty than can be specifically anticipated. 5g3f2b

NIC staff involved in producing the last two Catalogs have kept a diary of problems as encountered (summarized below). In the reading of this diary the impression of the staff is reinforced that problems of various kinds seem to occur serially: as soon as one problem is corrected, another is in line to appear. 5g3f2c

It is true that we could have issued typed versions of the NIC Catalog, the Resource Notebook, the User Guide, and the Directories in less time than it has taken to produce them as online files capable of being machine-updatable and printable on demand. But the trade-off always had to be made between service to the Network by getting out the information, and the benefit to be gained from experimentation with machine methods, eventually leading to a better product. 5g3f2d

DESIGN PROBLEMS 5g3f3

In designing the printed catalog, no existing catalog was taken as a model. Each alternative format which

offered advantages to the user and which was adaptable to our printer was considered. Selection of data elements for the printed catalog and for online searching was evolutionary, and extensive programming was necessary to accommodate changes in elements and format. 5g3f3a

The online catalog is tied to the printer format and is less readable than is desirable; we plan to put effort into design of an online version which is better adapted to display and teletype printout. 5g3f3b

MACHINE PROBLEMS 5g3f4

The occasional unreliability of the system seemed to adversely affect the Catalog operation more than other work at the site. Many times files containing programs or citations were lost in dumps or for other reasons. 5g3f4a

For some reason yet undetermined, large, heavily manipulated files have gone bad. A great number of files had to be reconstructed from earlier versions. 5g3f4b

An off-hours schedule is required because of the load placed on the system by catalog production, and consequently time was often lost in waiting for machine availability in off-periods. The process has been slow, consuming several hours of an evening, during which the operator had to keep an eye on the terminal. A late run sometimes had to be aborted because it ran into the dump time. 5g3f4c

Delays were caused by printer malfunction. Sometimes several days were lost because the printout for the reproducible master could not be obtained until the printer was cured of some aberrant behavior. 5g3f4d

Limitations of the line printer caused some compromises with an ideal design. We have experimented with various formats to achieve clarity if not beauty. 5g3f4e

PROGRAM PROBLEMS 5g3f5

The complexity of the present catalog production process of calling files and using programs led to

time-consuming mistakes. (The new catalog production processor should help greatly here). 5g3f5a

The continual improvement in analyzer and formatter programs required debugging and close examination of results. 5g3f5b

The continual evolution of NLS in general often has resulted in a new version on which some subsystem used in making a catalog would no longer run or run correctly. The process of catalog making with its large files, diversity of operations and long run times has proven a useful NLS bug finding tool. 5g3f5c

The programs for formatting the listings and indices were primitive at the beginning, and have been changed as the possibilities of the medium were explored. Each change in programs has meant the usual debugging. It has also meant extensive examination of the effects of each change on the citations resulting from the new manipulation of the data elements. 5g3f5d

The heretofore unreached limitations on the size of NLS files and fields have been brought out by the unusually large size and the unusually heavy machine operations required for formatting long bibliographic citations. 5g3f5e

INPUT TEXT PROBLEMS 5g3f6

The most obvious problem, and the most common, is mispelling, at the manual coding stage or at the typewriter input stage or by accident in making editing changes. 5g3f6a

Misunderstandings between staff members on file naming and other cataloging conventions often occurred during periods of new procedure development and staff training, all of which introduced delays or bad data or bad files. Most of these types of problems have been cured by brief weekly meetings of people involved with the various phases of catalog production, catalogers, coders, programmers,, etc. 5g3f6b

The selection of information from the document, in the

coding process, is vital to the retrievability of the information in the document, and errors in judgment in this selection must be caught to make the citation useful. 5g3f6c

The diversity of data, in type and length, of document citations causes it to be impossible to predict exactly what a formatting change will do to some citations. Trial and error are needed to help tailor input to the requirements of consistency necessary to produce an informative citation for the complete listing and for the on-line indices. 5g3f6d

REPRODUCTION PROBLEMS 5g3f7

To reduce the bulk of the Catalog listings and indices, the second issue of the catalog was formatted to squeeze as much information on a line and page as practical. The appearance of the final product is then dependent to some degree on type of offset system used and the proficiency and care of the reproduction department in photoreducing the masters. In some cases, the product has not been what we desired, because of the quality of our printout, or, more often, because of unnecessarily great reduction or incorrect photoprocess. On two occasions we have had to send the order back to be rerun. 5g3f7a

Photo reproduction is done centrally at SRI unless the delay would be insupportable; we regularly send the Catalog outside for repro, at an increased price and a still unsatisfactory schedule. 5g3f7b

COLLATION PROBLEMS 5g3f8

Errors in collation occur with predictable regularity but in unpredictable places in the document, of course. NIC is forced to do much of its own collation, and to check the collation done outside, with resultant delay. 5g3f8a

CONCLUSION 5g3f9

Familiarity with other centers building machine-produced catalogs, (see References Section 2c)

has convinced us that the above types of problems are part of the present state of the art of the information business and that any installation planning to do these types of operations should plan on a shakedown period to work them out. If their system, like ours, is constantly evolving as part of planned research and development, this shakedown period may always exist. 5g3f9a

SOME FUTURE PLANS 5g3g

At the present time, dialog items submitted online to the Journal and mailed to us for distribution offline, and more formal documents such as reports, are intermixed in our catalog. As the collection grows these classes of items will be separated to maintain ease of catalog browsing offline and online. 5g3g1

We will also probably produce the catalog in book form as now, and machine-produced cards can be distributed to the stations with each item to aid Station Agents in maintaining an up-to-date catalog of their local collections. 5g3g2

Plans exist to consider in the next year or two distribution of items to site collections on microfilm. 5g3g3

The Directory of Network Participants 5g4

The Directory of Network Participants is automatically produced from information in the identification file described earlier. The Directory contains several views of the information in this file. There are three main categories of records in this file: individuals, dialog groups, and affiliations. Affiliations are organizations and are special cases of dialog groups. The Directory contains a comprehensive online listing of IDENTS and names of all items in the file, brief and extended listings of individuals, dialog groups, and affiliations, a listing of principle investigators associated with each network site and ARPA contract, liaison, station agents, and special mailing lists (the latter are special cases of dialog groups). Examples of Directory formats are given. 5g4a

SRI-ARC 8 JUNE 1972 13041
Network Information Center
Development and Operations
Offline Services

COMPREHENSIVE LIST OF IDENTS:

5g4b

DF	David Farber (UCI)
DGB	Daniel G. Bobrow(BBN-TENEX)
DHL	Duncan H. Lawrie(ILL)DIA
Don I.	Andrews(SRI-ARC)
DL	Don Limuti (SRI-ARC)
DLM	Dan L. Murphy (BBN-TENEX)
DLM	Donal L. McNally (Tinker)
DLS	Duane L. Stone (RADC)
DLS2	Daniel L. Slotnick (ILL)
DM	Donald McCracken (CMU)

BRIEF LIST OF AFFILIATIONS:

5g4c

BBN-TENEX	B. B. and N. - TENEX Group Bolt Beranek and Newman Inc. 50 Moulton Street Cambridge, Massachusetts 02138
CASE	Case Western Reserve University 10900 Euclid Avenue Room 306, Crawford Hall Cleveland, Ohio 44106
CCA	Computer Corporation of America 565 Technology Square Cambridge, Massachusetts 02139
CCCTF	Canadian Computer Communications Task Force 100 Metcalfe Street 4th Floor Ottawa 2, CANADA

CHIU University of Chicago
Institute for Computer Research
University of Chicago
Chicago, Illinois 60637

LINC-67 M.I.T. Lincoln Lab - 67 Group
244 Wood Street
Lexington, Massachusetts 02173

EXTENDED LIST OF AFFILIATIONS:

5g4d

BBN-TENEX B. B. and N. - TENEX Group
Bolt Beranek and Newman Inc.
50 Moulton Street
Cambridge, Massachusetts 02138

DGB	Daniel G. Bobrow	(617)891-1850 ext 330 ←	Principal Investigator
SCB	Stephen C. Butterfield	(617)891-1850 ext 419	
SUC	Steven C. Chipman	(617)891-1850 ext 358 ←	Station Agent
REK2	Robert E. Kahn	(617)891-1850 ext 340	
JM	John Makhoul	(617)891-1850 ext 234	
MM	Mac McKinley	(617)891-1850 ext 351 ←	Laison

SRI-ARC 8 JUNE 1972 13041
Network Information Center
Development and Operations
Offline Services

EXTENDED LIST OF GROUPS:

5g4e

BBN-TENEX	Dan L. Murphy (DLM) Bolt Beranek and Newman Inc. Computer Science Division 50 Moulton Street Cambridge, Massachusetts 02138	(617) 491-1850 ext 351
CASE	Patrick W. Foulk (PWF) Case Western Reserve University Computing and Information Sciences 10900 Euclid Avenue Cleveland, Ohio 44106	(216) 368-2936
CCA	Richard A. Winter (RAW) Computer Corporation of America 565 Technology Square Cambridge, Massachusetts 02139	(617) 491-3670

BRIEF LIST OF INDIVIDUALS:

5g4f

Murphy, Dan L. (DLM)	BBN-TENEX	(617)491-1850 ext 351
Naficy, Hamid (HN)	UCLA-NMC	(213)825-2377
Naylor, William E. (WEN)	UCLA-NMC	(213)825-2012 ext 2368
Nelson, Lou C. (LCN)	UCLA-NMC	(213)825-4733 or 825-2368
Newell, Allen (AN)	CMU	(412)621-6200 ext 151
North, Jeanne B. (JBN)	SRI-ARC	(415)326-6200 ext 4119
Norton, James C. (JCN)	SRI-ARC	(415)326-6200 ext 2124
O'Sullivan, Thomas (TO)	RAY	(617)762-6700 ext 2120

EXTENDED LIST OF INDIVIDUALS: 5g4g

Dan L. Murphy (DLM) BBN-TENEX (617)491-1850 ext 351 Murphy
Bolt Beranek and Newman Inc.
Computer Science Division
50 Moulton Street
Cambridge, Massachusetts 02138

Hamid Naficy (HN) UCLA-NMC (213)825-2377 Naficy
UCLA - Network Measurement Center
Computer Science Department
3732 Boelter Hall
Los Angeles, California 90024

William E. Naylor (WEN) UCLA-NMC (213)825-2012 ext 2368 Naylor
UCLA - Network Measurement Center
Computer Science Department
3732 Boelter Hall
Los Angeles, California 90024

We plan in the future to expand the information in the Directory to include additional information of value to the ARPANET community such as individuals' research interests, description of functions of each dialog group, etc. 5g4h

ARPA Network Resource Notebook 5g5

For people to be able to effectively utilize the resources of the network, they must know what resources are available. The initial service to meet this need is the ARPA Network Resource Notebook. This Functional Document was launched in 1971 jointly by BBN and NIC. BBN designed the initial information format, collected initial entries from each site, and did additional editorial work to insure uniformity. Responsibility for distribution and maintenance was handled by the NIC. We also transcribed the material and made it available online. 5g5a

As the number of sites grew it became clear that it should have its information content expanded, with more specialized sections on specific classes of resource, and that it needed an index. 5g5b

An index was prepared at the NIC and as we wanted to develop more automatic aids to producing special views of the resource information, and as BBN felt they had fulfilled their initial obligation, future editorial responsibility for the Resource Notebook has passed to the NIC. 5g5c

ARPA Network Current Network Protocols 5g6

Successful intercomputer communication over the ARPANET depends on the development and implementation of various classes of communication protocols. To make this information widely available and easily usable in a form people could tell was up to date, the various protocols were collected together as one Functional Document and are maintained as such by the NIC. This has proven to be a simple, but useful service. 5g6a

Network Information Center User Guide 5g7

To enable users to learn and use NLS, the Journal and other online NIC services, an extensive User Guide was prepared covering that subset of TNLS functions felt to be initially useful to network users. Because NLS is a constantly evolving system and such a document as the User Guide would be expensive to constantly change and reproduce, we keep up-to-date a document we call Folklore which contains sections corresponding to sections in the User Guide which note changes to the system, errors in the User Guide, helpful suggestions, usage, warnings, etc. This document is distributed periodically and is available online. 5g7a

We are presently writing a Primer containing a highly restricted subset of TNLS, but which is sufficient for people to create and study documents and use the Journal capabilities. 5g7b

A number of changes to the NLS command language are under study. When these have been settled on, a new version of the User Guide will be written. 5g7c

Support of Offline Network Dialog 5g8

At the present time the NIC supports several dialog groups, the main ones being the Network Working Group (NWG) of 68 members and its subgroups, the Speech Understanding Research Group (SUR) of 20 members, and the Computer Based Instruction Group (CBI) of 25 members. These groups can use our online facilities or mail us a copy of a document and indicate that they want it distributed to a named

group. We duplicate, catalog, and distribute the document to the appropriate group of individuals and station collections. This mode of offline and online operation has been a well received and used service to aid creation of a sense of community in the particular distributed groups and as an aid to their collaborative functioning. 5g8a

Requests for Network Information 5g9

People interested in the ARPANET, from organizaaons not directly connected with the network, from organizations connected with the network but not familiar with the functioning of the network, and people in various media services, frequently contact us for specific documents or general background information about the network. We supply information both verbally over the telephone and in person, and send appropriate introductory or other material as required to meet these requests. We frequently refer people to someone within the network community for additional or more detailed information when appropriate. 5g9a

We feel as part of the NIC service it would be desirable to produce at NIC more descriptive information about activities in the ARPANET community. Because of staff and budget limitations we have not been able to take on this role as yet and have depended on people within the community to write this type of documentation. Not nearly enough has been written, however, and we hope in the coming year to try to some degree to fill this network documentation gap. 5g9b

TRAINING 5g10

To launch the online use of NIC services, we have run at SRI 6 two-day training courses in the use of NLS and the Journal and we ran one course at MIT-DMCG using their computer and the ARPANET to contact our system for training. These courses have been attended by one or more people from the sites with online access to NIC services and by people interested in the network and desiring a feel for network usage and background on present network operation. The size of each class has generally run around 12 people, although over 20 attended the class given at MIT. 5g10a

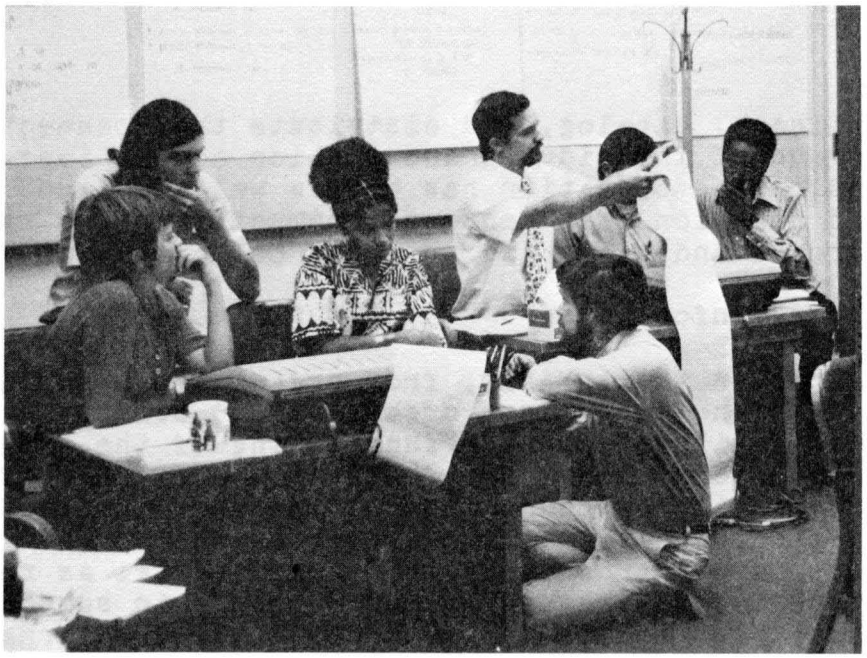


FIGURE 22. Instructors (right in each group) work with students during practice period in NIC TNLS training course. Class alternates between brief lectures and practice periods. NIC provides a terminal for each student or for every two students depending on class size.

5g10a1

These people have returned to their sites to use our system and help others at their site learn to use it as well. 5g10b

We have been constantly evaluating how best to teach the use of NLS and have been getting useful feedback on areas of improvement needed in TNLS to make it easier to learn and use. 5g10c

To provide a useful service to a distributed community requires more active on-site user instruction and help than we have yet been able to provide. We hope in the coming year to be able to free resources to provide more of this kind of assistance. 5g10d

As part of this teaching process a series of "workbooks" which take a person through commonly used TNLS operations has been prepared. 5g10e

We expect to continue evolution of our teaching aids and NLS features to make the system responsive to both the needs of inexperienced and experienced users. We hope also to add those features which could make NLS more self-instructive. 5g10f

EXPERIENCE USING THE ARPANET 5h

Experience using the ARPANET has proven quite satisfactory at this stage of its development. The ARPANET really only came alive in mid-October 1971 even though the central communications network had been operational over a year earlier, because it was only then that more than one or two hosts were operational on the network with the needed protocols. 5h1

The ARPANET, the network interfaces, local hosts, and network software comprise a very large, sophisticated system in which there are many places where breakdowns can occur. The central network facility has been quite reliable, with most of the breakdowns and bugs in local hosts and network software. With the many possible places of breakdown, early network users and servers have had to be patient and understanding of their own local system and those serving them on the network. Reliability of hosts on the network seems to be constantly improving. 5h2

Response to users over the network seems to generally be quite good in spite of having two computer systems with their layers of network protocol software in the loop. 5h3

It is clear that there is much yet to be learned about handling network protocols in local operating systems and in network hardware development. 5h4

We are, however, quite pleased with directions of network development and are convinced that this type of technology is here to stay and will have a significant impact on the development of this project (ARC), the nature of the organization of research and development generally, and information technology. 5h5

CONCLUSION 5i

The Network Information Center is, we believe, an example of a new type of information service which has significant future potential and, even though it is presently in an experimental and development phase, is providing useful online and offline services to the ARPANET community. Now that a basic operational service is in existence, future attention will be turned not only to further evolution of the range of services offered, and the quality of each service, but also to an analysis of the costs of each service. The cost of information services is a topic of much discussion in the literature but one on which there is little concrete data (see Reference Section 2b). We hope in future reports to be able to describe in some detail the costs of various NIC services and

compare them with the reported costs of similar activities at other centers or performed by other means. 5i1

We also hope to study the way various NIC services are being used by NIC clientele and to evaluate the utility of each service in more detail. 5i2

REFERENCES 5j

(5145,) Current Catalog of the NIC Collection. ARPA Network Information Center, Stanford Research Institute, Menlo Park, California. (Current version dated 1 February 1972.) Separately paged. 5j1

(5417,) Jim White (UCSB). NWG/RFC 74, Specifications for Network Use of the UCSB On-Line System. ARPA Network Information Center, Stanford Research Institute, Menlo Park, California 94025. 16 October 1970. 1p. 5j2

(5621,) R. Anderson, E. Harslem, J. Heafner (RAND). NWG/RFC 83, Language-Machine for Data Reconfiguration. ARPA Network Information Center, Stanford Research Institute, Menlo Park, California 94025. 18 December 1970. 12p. 5j3

(5740,) John T. Melvin, Richard W. Watson (SRI-ARC). NWG/RFC 97. A First Cut At a Proposed Telnet Protocol. ARPA Network Information Center, Stanford Research Institute, Menlo Park, California 94025. 15 February 1971. 1p. 5j4

(5772,) John Heafner, Eric Harslem (RAND). Working Note on Data Reconfiguration Language/Service. ARPA Network Information Center, Stanford Research Institute, Menlo Park, California 94025. 26 February 1971. 23p. 5j5

(5775,) James E. White (UCSB). NWG/RFC 105, Network Specifications for Remote Job Entry and Remote Job Retrieval at UCSB. ARPA Network Information Center, Stanford Research Institute, Menlo Park, California 94025. 22 March 1971. 8p. 5j6

(6025,) Peggy Karp. Bibliography of Literature on Computer Networking. MITRE Corporation, McLean, Virginia. December 1970. 87p. 5j7

(6740,) ARPA Network Resource Notebook. ARPA Network Information Center, Stanford Research Institute, Menlo Park, California. (Current version dated 22 November 1971.) Separately paged. 5j8

(6808,) UCSB System 360/75, UCSB Computer Center. University of California at Santa Barbara, Computer Sciences Department, Santa

Barbara, California. In ARPA Network Resource Notebook (6740,), p.3. April 1971. 5j9

(7104,) ARPA Network Current Network Protocols. ARPA Network Information Center, Stanford Research Institute, Menlo Park, California 94025. (Current version dated 12 August 1971.) Separately paged. 5j10

(7120,) Jeanne North (SRI-ARC). Functional Documents and Their Revision. Augmentation Research Center, Stanford Research Institute, Menlo Park, California 94025. 23 June 1971. 1p. 5j11

(7135,) Peter Deutsch (PARC). NWG/RFC 190, DEC PDP-10 -- IMLAC Communication System. ARPA Network Information Center, Stanford Research Institute, Menlo Park, California 94025. 13 July 1971. 15p. 5j12

(7136,) Charles Irby (SRI-ARC). NWG/RFC 191, Graphics Implementation and Conceptualization at ARC. ARPA Network Information Center, Stanford Research Institute, Menlo Park, California 94025. 13 July 1971. 4p. 5j13

(7137,) Richard W. Watson (SRI-ARC). NWG/RFC 192, Some Factors Which a Network Graphics Protocol Must Consider. ARPA Network Information Center, Stanford Research Institute, Menlo Park, California 94025. 12 July 1971. 22p. 5j14

(7141,) Richard W. Watson (SRI-ARC). NWG/RFC 196, A Mail Box Protocol. ARPA Network Information Center, Stanford Research Institute, Menlo Park, California 94025. 20 July 1971. 4p. 5j15

(7470,) TNLS User Guide: Preface, Syntax and Contents. ARPA Network Information Center, Stanford Research Institute, Menlo Park, California 94025. (Current version dated 1 September 1971.) Separately paged. 5j16

(7612,) Richard W. Watson (SRI-ARC). NWG/RFC 221, A Mail Box Protocol, Version 2. ARPA Network Information Center, Stanford Research Institute, Menlo Park, California 94025. 25 August 1971. 5p. 5j17

(7834,) James C. Norton (SRI-ARC). Outline for Establishing People Support Team (PSST) at ARC. Augmentation Research Center, Stanford Research Institute, Menlo Park, California 94025. 2 November 1971. 11p. 5j18

(8056,) Abhay Bhushan (MIT-DMCG), Bob Braden (UCLA-CCN), Eric Harslem, John Heafner (RAND), Alex McKenzie (BBN-NET), John Melvin (SRI-ARC), Bob Sundberg (HARV), Dick Watson (SRI-ARC), Jim White (UCSB). NWG/RFC 278, Revision of the Mail Box Protocol. ARPA Network Information Center,

Stanford Research Institute, Menlo Park, California 94025. 17 November 1971. 4p. 5j19

(8158,) Richard Watson (SRI-ARC). Summary of 1971 Activities, ARPA Contractors' Meeting, 9-10 December 1971. ARPA Network Information Center, Stanford Research Institute, Menlo Park, California 94025. 9 December 1971. Pages not given. 5j20

(8289,) Richard W. Watson (SRI-ARC). Reply To JBL on Output Device Teletype. ARPA Network Information Center, Stanford Research Institute, Menlo park, California 94025. 20 December 1971. Pages not given. 5j21

(8295,) Richard W. Watson (SRI-ARC). NWG/RFC 289, What We Hope Is An Official List of Host Names. ARPA Network Information Center, Stanford Research Institute, Menlo Park, California 94025. 21 December 1971. 3p. 5j22

(9474,) Donald E. Walker (SRI). Interactive Bibliographic Search: The User/Computer Interface, Proceedings of a Workshop. AFIPS Press, Montvale, New Jersey. 1971. 311p. 5j23

(9868,) J. B. North (SRI-ARC). Codes Used in the Master Catalog. Augmentation Research Center, Stanford Research Institute, Menlo Park, California 94025. January 1972. 6p. 5j24

(9934,) DEX User Guide. Augmentation Research Center, Stanford Research Institute, Menlo Park, California 94025. 19 June 1972. 61p. 5j25

NETWORK PARTICIPATION

Richard W Watson John T Melvin
Charles H Irby and James E White 6

INTRODUCTION 6a

During the prime period of this report, general network participation other than direct Network Information Center activities has been active. 6a1

Our network participation activity has been in two main areas, protocol development through work in several protocol design committees and general network coordination through membership on the short lived Network Working Group Steering Committee and its successor, the Network Facilitators Group. 6a2

PROTOCOL DEVELOPMENT 6b

We helped launch the Telnet Protocol design committee at the February 1971 Network Working-Group (NWG) meeting with the document "A First Cut at a Proposed Telnet Protocol," RFC 97, NIC 5740 and participated actively in the design of the protocol. The Telnet Protocol allows user typewriter terminals of various types and attached to the users host to communicate with serving hosts through definition of a standard Network Virtual Terminal System. The Telnet Protocol is described in "ARPA Network Current Network Protocols", NIC 7104. Dick Watson and John Melvin were active in this area. 6b1

At the May 1971 NWG meeting we helped launch the design committee set up to study the problem of general network data and file transfer. Two initial protocols were designed at that meeting, one for data transfer and another for file transfer. Although it was felt at the time that further work and experimentation was needed on this data and file transfer problem, the resulting protocols were felt to be adequate to gain initial experience. These Protocols are documented in "ARPA Network Current Network Protocols" NIC 7104. 6b2

Because most sites were preoccupied with implementing their Network Control Programs (NCP)and Telnet Protocols, implementation at a few sites, including SRI-ARC, of the Data and File Transfer Protocols did not start until early 1972. At this time early implementation experience and further experience in using the network indicated that the design of the Data and File Transfer Protocols should be reconsidered. A meeting of the design committee was held at MIT in April 1972 which resulted in a new design. The results of this work are presently being documented by Abhay Bhushan of MIT-DMCG. 6b3

SRI-ARC 8 JUNE 1972 13041
Network Participation
Protocol Development

One of the services of the NIC is to facilitate network dialog by use of the Dialog Support System (DSS) of SRI-ARC's Online System (NLS). To deliver documents and messages entered into the DSS through the network to printer files at remote sites is simplified if a protocol built on the File Transfer Protocol is supported by each site. A protocol called the "Mailbox" Protocol was proposed in RFC 196, NIC 7141 and revised in RFC 221, NIC 7612 and RFC 278, 8056. The new File Transfer Protocol being developed may include a "mailbox" capability (i.e., a capability to deliver printer files to remote sites) and therefore the future of the Mailbox Protocol is not certain at this point. Dick Watson, John Melvin and Jim White have been active in the above areas. 6b4

In July 1971, the first meeting of the Network Graphics Group (NGG) was held to discuss requirements for a protocol to handle interactive graphics over the network. One of our goals in the graphics area is to support the display version of NLS over the network. We have been working with L.P. Deutsch of Xerox, Palo Alto Research Center in this area, as Xerox wishes to use NLS from an IMLAC display. The initial work here was described by Deutsch in "DEC PDP-10--IMLAC Communication System," RFC 190, NIC 7135 and by Irby in "Graphics Implementation and Conceptualization at ARC", RFC 191, NIC 7136 At the same time "Some Factors which a Network Graphics Protocol Must Consider," RFC 192, NIC 7137, was published by Watson. 6b5

Further meetings of the NGG have been held, with the most recent in April 1972, at which we participated in the design of an initial experimental interactive graphics protocol. 6b6

We have been experimenting with running the display version of NLS over the network from two sites, BBN and UCLA-NMC. The results are encouraging from BBN, which has a 9600-baud interface to their IMLAC and a resident NCP. The UCLA-NMC system runs its Telnet and its NCP as user programs. They have a 1200 baud connection to their IMLAC. When loaded, response is considerably slower than experienced locally at SRI or experienced by BBN. When both the UCLA-NMC and SRI-ARC machines are lightly loaded, response at UCLA is barely satisfactory. 6b7

Charles Irby has been active in the network graphics area. 6b8

Through Jim White, who joined SRI-ARC in early April, we have been participating in the design of a Network Remote Job Entry Protocol. The most recent design meeting was held in April 1792. 6b9

Some work has been done by the NWG on considering the requirement for a Network Data Management Protocol that would support management of distributed data. We have maintained contact with the people working in

this area, but other than defining some of our needs for reporting to this group we have not been active participants as yet. We expect to work more actively in this area. 6b10

Plans for the future call for continued active participation in the protocol design areas mentioned above and for implementation of those protocols for experimental and normal usage, as appropriate. 6b11

NETWORK COORDINATION 6c

A Network Working Group Steering Committee was set up at the May 1971 NWG meeting of which John Melvin was a member. This group planned the October 1971 NWG meeting and then was replaced by a Network Facilitators Group consisting of nine members geographically distributed. This group has as its purpose to help give detailed technical information and personal assistance to people desiring to get on the network or make technical contact with the network community. This group has also served a trouble shooting role in general network coordination. John Melvin and Jim White are members of this group. John left SRI to work at RAND in April 1972. 6c1

REFERENCES 6d

(5740,) John T. Melvin, Richard W. Watson (SRI-ARC). NWG/RFC 97. A First Cut At a Proposed Telnet Protocol. ARPA Network Information Center, Stanford Research Institute, Menlo Park, California 94025. 15 February 1971. 1p. 6d1

(7104,) ARPA Network Current Network Protocols. ARPA Network Information Center, Stanford Research Institute, Menlo Park, California 94025. (Current version dated 12 August 1971.) Separately paged. 6d2

(7135,) Peter Deutsch (PARC). NWG/RFC 190, DEC PDP-10 -- IMLAC Communication System. ARPA Network Information Center, Stanford Research Institute, Menlo Park, California 94025. 13 July 1971. 15p. 6d3

(7136,) Charles Irby (SRI-ARC). NWG/RFC 191, Graphics Implementation and Conceptualization at ARC. ARPA Network Information Center, Stanford Research Institute, Menlo Park, California 94025. 13 July 1971. 4p. 6d4

(7137,) Richard W. Watson (SRI-ARC). NWG/RFC 192, Some Factors which a Network Graphics Protocol Must Consider. ARPA Network Information Center, Stanford Research Institute, Menlo Park, California 94025. 12 July 1971. 22p. 6d5

(7141,) Richard W. Watson (SRI-ARC). NWG/RFC 196, A Mail Box Protocol.

ARPA Network Information Center, Stanford Research Institute, Menlo Park, California 94025. 20 July 1971. 4p. 6d6

(7612,) Richard W. Watson (SRI-ARC). NWG/RFC 221, A Mail Box Protocol, Version 2. ARPA Network Information Center, Stanford Research Institute, Menlo Park, California 94025. 25 August 1971. 5p. 6d7

(8056,) Abhay Bhushan (MIT-DMCG), Bob Braden (UCLA-CCN), Eric Harslem, John Heafner (RAND), Alex McKenzie (BBN-NET), John Melvin (SRI-ARC), Bob Sundberg (HARV), Dick Watson (SRI-ARC), Jim White (UCSB). NWG/RFC 278, Revision of the Mail Box Protocol. ARPA Network Information Center, Stanford Research Institute, Menlo Park, California 94025. 17 November 1971. 4p. 6d8

COMPUTER FACILITY

by Donald C Wallace, James C Norton,
John T Melvin, Donald I Andrews, Charles H Irby,
Edwin K Van De Reit, and Kenneth E Victor 7

HARDWARE 7a

Introduction: 7a1

At the end of the first year of this contract, we transferred our computer operations from an XDS-940 to a PDP-10 computer. The transfer effort is described in our interim report for the first year (8277,). 7a1a

Hardware activity during the past year has focused on additional tuning of the new configuration, maintenance, troubleshooting and operation of the facility, and some upgrading of critical parts of the system. 7a1b

FIGURE 23. (Opposite) ARC Computer System. 7a1b1

Present Configuration 7a2

The present ARC computer facility configuration is as follows: 7a2a

Digital Equipment Corporation (DEC) equipment is the heart of our facility, providing the computer, core memory, and mass storage devices (discs, magnetic tape units). 7a2a1

PDP-10 7a2a1a

The KA10 Central Processor has a 36-bit word length and an 18-bit address field. It controls computer cycles, executes machine-language instructions, and handles priority interrupts. It interfaces with the outside world through its I/O Bus and Memory Bus. 7a2a1a1

Processor 7a2a1b

Memory - 8 MA10's (2 ME10's being added in June 1972) 7a2a1c

These are ferrite core memories and are used with the KA10 processor. The memory allows for storage of 37-bit words (36 bits and parity) and has a 1 us MA10/ME10 cycle time. Each memory box has a storage capacity of 16,384 words. 7a2a1c1

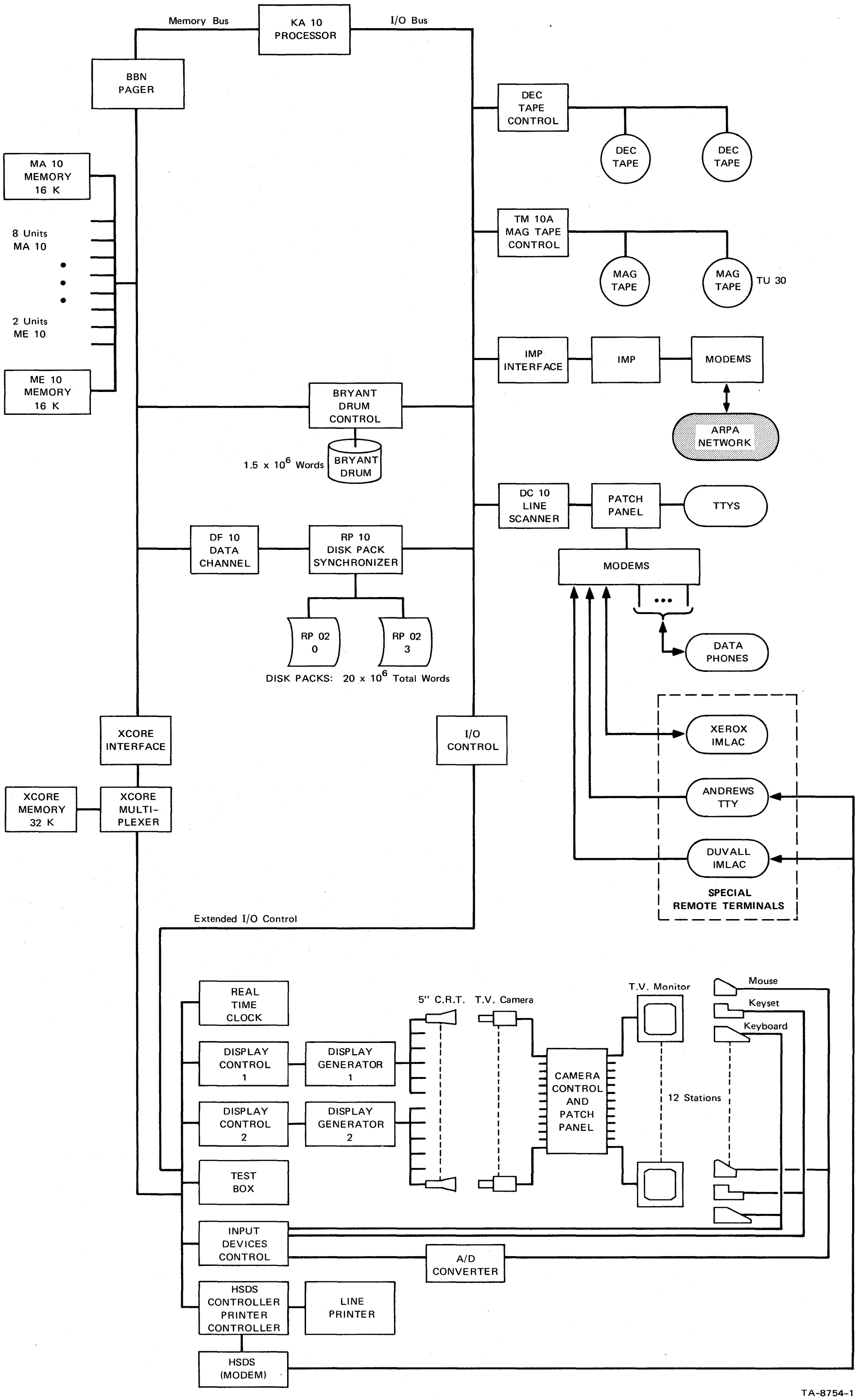


FIGURE 23 AUGMENTATION RESEARCH CENTER-COMPUTER SYSTEM CONFIGURATION

Memory Interface DF-10 7a2a1d

The DF10 Data Channel is a high-speed trapsfer device (10↑6 words/sec). In the ARC configuration, it accomplishes direct data transfer between the (RP02's) and memory. 7a2a1d1

Once enabled, data transfers independently of the program in progress, thereby releasing the central processor for other operations. 7a2a1d1a

Disc Packs - RP-02's 7a2a1e

Our RP02's each provide storage for 5,196,800 36-bit words. Average access time is 62.5 ms. The transfer rate is 15 ms/word. The ARC system has four online RP02's for a total storage of about 20 x 10↑6 words. An additional RP02 is available to back up the disc system. 7a2a1e1

Disc Pack Controller - RP10 7a2a1f

The RP10 provides the interface logic between the DF10 Data Channel and the RP02 Disc Pack Driver. 7a2a1f1

DEC tape Units 7a2a1g

These are special magnetic tape units used for loading programs into the core memory. They are usually used for bringing up the ARC system. 7a2a1g1

Mag Tape Units and Controller 7a2a1h

These units enable mass storage of information onto magnetic tape and are used in performing disc dumps and for file archival processes. 7a2a1h1

Line Scanner 7a2a1i

The DC10 Data Line Scanner provides a timesharing two-way interfage between the PDP-10 central processor and a maximum of 64 teletype-like stations. The current configuration handles 24 stations. 7a2a1i1

Bolt, Beranek, and Newman, Inc. (BBN) has provided much of the special hardware and software that modifies the standard PDP-10 system to make it compatible with ARC and Network requirements. 7a2a2

Paging Box 7a2a2a

This device interfaces the PDP-10 central processor to the core memories. It facilitates the swapping of pages (512 36-bit words) between the core memories and either the drum or the disc. 7a2a2a1

Interface Message Processor (IMP) and IMP Interface 7a2a2b

The IMP is the interface between the ARC Network Information Center and the ARPA Network. It connects to the PDP-10 via the I/O Bus and connects to the rest of the Network via telephone lines. 7a2a2b1

Bryant Drum and Interface 7a2a3

The Bryant drum is a mass storage device with a capacity of 1,566,720 words and an average access time of 16ms. Once enabled, data transfer with the core memories proceeds independently of the programs in progress, thereby releasing the central processor for other operations. The drum is the primary transfer (swapping) device to the core memories. 7a2a3a

Bryant Disk 7a2a4

This mass storage device has a capacity of 23 x 10↑6 36-bit words. At present it is used as backup for the DEC RP02's while further uses for it are being considered. 7a2a4a

I/O Control Box 7a2a5

This device is used to extend the PDP-10 I/O Bus. It also provides manual control over the peripheral devices it interfaces. 7a2a5a

External Core (Xcore) 7a2a6

This is a 32k 24-bit memory. It is now used for storage of display and keyboard information and other non-critical information transferred at slower rates. 7a2a6a

Xcore Multiplexer 7a2a7

This attaches 8 ports to Xcore. (The main core MA10/ME10's have 4 ports as part of their structure). 7a2a7a

Xcore Interface Box 7a2a8

This device provides for the proper timing and voltage interface between the PDP-10 memory and the Xcore Multiplexer. 7a2a8a

Real Time Clock 7a2a9

This clock provides the reference for all times recorded by the system. 7a2a9a

TTY Patch Panel 7a2a10

This connecting panel allows some of the many TTY and Modem inputs to be connected to the 24 channels available on the line scanner. 7a2a10a

Dataphones 7a2a11

There are 8 Dataphones and Modems connected to the system. 7a2a11a

Display Controllers, Tasker Display Generators, and Closed Circuit TV 7a2a12

These devices enable local users to view any of the 12 television monitors (located at the display consoles). These monitors display information stored by the system in Xcore. 7a2a12a

Data Products Line Printer 7a2a13

This device provides for hardcopy printout of user and system files. 7a2a13a

Test Box 7a2a14

This unit occupies a part of Xcore and is used as a troubleshooting tool. 7a2a14a

Input Devices Controller (IDC) 7a2a15

This equipment handles information from the display consoles (12 keyboards, keysets, and mice) and stores it in Xcore to await processing. 7a2a15a

A/D Converter 7a2a16

This converts analog mouse coordinates to digital coordinates to be stored in Xcore via IDC. 7a2a16a

Display Consoles (12, plus spares) 7a2a17

These each are composed of: 7a2a17a

Mouse
Keyset
Keyboard
Video Displays 7a2a17a1

Console Patch Panel 7a2a18

This enables video output from the display system to be interchanged among the various display consoles within the ARC work area and is also useful in troubleshooting. 7a2a18a

Illustration to be used: 7a2b

ARC PDP-10 System Configuration Layout 7a2b1

Problems We Have Been Facing 7a3

ARC Service Problems 7a3a

One of ARC's key objectives is to provide reliable service to its augmentation system users at as reasonable a cost level as we can within the context of our part-developmental, part-service environment. 7a3a1

We have provided many ARC and Network users with NLS service during the past year. In this period, ARC and Network users have in many instances experienced system accessability and user-response at what we consider to be undesirable levels. 7a3a1a

The main cause of such lowered service levels has been problems with our hardware, although some software problems have also been encountered. 7a3a1b

We have been concentrating on the various hardware problems that have caused lower-than-desired service levels. 7a3a2

One source of trouble has been the external core (Xcore) configuration through which we have run the Network Interface, the ARC displays, the line printer, and other devices. 7a3a2a

Sensitive cable connections that have broken with handling during maintenance and trouble-shooting work, many cards that have failed, and basic internal Xcore grounding design flaws were the main causes of Xcore failures. 7a3a2a1

New cables have been made and are being installed. A different grounding scheme has been implemented and appears to have reduced the noise levels previously experienced in Xcore. 7a3a2a1a

A new BBN Network Interface has been installed that does not connect to the system through our Xcore. In the future, Xcore trouble will not bring down our Network connection (at least not as in the past). 7a3a2b

The DEC PDP-10 has had several failures in the past few months. DEC responds quickly and effectively to such occurences. 7a3a3

Any timesharing system is susceptible to such failures. The fact that we have only one machine puts us in a position of being more vulnerable to service interruption than that experienced by large commercial utilities, where there are many machines. When one machine goes down, another is switched in, frequently without the users seeing much effect. 7a3a3a

We plan to discuss with commercial utilities the possibility of their providing all or part of another machine for the NIC portion of our NLS service in an effort to: 7a3a3b

1. Become more reliable and 7a3a3b1

2. To provide more computer service to NIC users. 7a3a3b2

We would have to secure additional funding for such an arrangement. 7a3a3c

The Bryant drum has failed several times recently, even with frequent maintenance by Bryant. The UNIVAC drums have been used as backup, but have been unreliable at times. 7a3a4

The UNIVAC drums are too expensive to keep as backup and don't speed the system significantly when used with the Bryant drums simultaneously. We have terminated the lease of the UNIVACs for cost reasons. 7a3a4a

We have tried swapping off a single DEC diskpack to see if we

can backup the Bryant equipment satisfactorily with that arrangement. 7a3a4b

The number of users supportable is considerably reduced, but service to a limited number (4 or 5 users) is satisfactory. 7a3a4b1

With the added 32k DEC memory (coming in June 1972), there will be less swapping, so that swapping off the diskpacks is expected to support more users than during early trials. 7a3a4b2

In addition, we plan to develop software that will swap off several of the packs, not just one. We also are contemplating adding another diskpack controller. 7a3a4c

If swapping off the diskpacks doesn't appear to be the best way to provide backup for the Bryant drum, we may have to add another Bryant drum. 7a3a4d

The Tasker display system is now five years old and is requiring an increasing level of maintenance and troubleshooting. 7a3a5

Since it provides the primary display facility to ARC DNLS users, its early replacement appears necessary, both to provide more reliable service and to upgrade the quality of the displays to current state-of-the-art performance levels. 7a3a5a

We have been actively trying to secure more of our equipment from commercial sources, while putting effort into making hardware that is unique-to-ARC more reliable through upgrading efforts. 7a3a6

The Xcore configuration and some interface hardware are still one-of-a-kind prototype equipment, and in some ways do not have the solid commercially-produced characteristics we now need. 7a3a6a

The Bryant disc is not being used now, since it was the source of serious reliability problems (crashing the system frequently) last year. Its functions are now being performed by the new DEC diskpacks. 7a3a6b

The Bryant disc is several years old now and is due for a major overhaul if it is to be further utilized in the ARC system. We have been looking for ways to use its capacity that do not put it in the mainline of our system operation. 7a3a6b1

SYSTEM SOFTWARE 7b

Imlac support for DNLS 7b1

A program written (by Peter Deutsch of Xerox Palo Alto Research Center -- XPARC) for an IMLAC display and processor and some modification to the display support monitor calls allowed us to offer Display NLS support over phone lines and through the ARPA Network. To date DNLS has been experimentally used by a remote ARC employee (about 100 miles away, phone line connection), by XPARC personnel (also phone line), at the Network Measurement Center at UCLA (ARPA NET), and at BBN (ARPANET) 7b1a

TENEX 7b2

In our inital use of BBN-TENEX the main concern was to just "make it work". In getting TENEX to run on our unique hardware configuration we made many extensive modififations and additions. In the ensuing year and a half of experience with TENEX and its evolution we have learned much. With the responibility of providing reliable computer resources for the NIC many heretofore overlooked requirements in running our facilities have become considerably more critical. 7b2a

It has become increasingly important to run as unmodified a version from BBN's distributed version as possible. Modifications are only made when there is a real user need and BBN cannot or will not make the appropriate mod or addition. When we do make additions or mods to TENEX we first attempt to implement them in a manner that would be of general use to TENEX users and secondly we notify BBN of the change in the hope that it will become a part of standard TENEX with a subsequent release. 7b2b

The following is an overview of many of the changes we have made here at SRI-ARC to BBN distributed TENEX. 7b2c

Further details, including implementation details, are available for the asking. 7b2c1

OPERATIONAL PROCEDURE CHANGES 7b2d

CHANGES TO START-UP AND RESTART PROCEDURES OF THE MONITOR 7b2d1

DTBOOT 7b2d1a

We have switched from using TENDMP (for loading the monitor from DECTAP) to using DTBOOT, a DEC-provided replacement for

TENDMP that is much easier to use and a better program than TENDMP. 7b2d1a1

Novice startup procedures 7b2d1b

We have changed the starting address of the monitor from 100 (which goes immediately to DDT) to SYSG01. 7b2d1b1

Thus the procedure necessary for a novice to bring up the system is easier and is as follows: 7b2d1b1a

readin DTBOOT

type CR.

One of the by-products of using DTBOOT is that it has built in default file names.

We have renamed the resident monitor to be SYSTEM.SAV

SYSTEM.SAV is the default name for loading for DTBOOT

DDT Flushing 7b2d1c

Several new flags were added to the monitor to control the use of memory for DDT. Basically three options are available to the system programmer. 7b2d1c1

The system may be run: 7b2d1c1a

Without DDT or the monitor symbol table.

With DDT, but no symbols

With both DDT and the monitor symbol table

Monitor routines are also provided to dynamically alter the state of DDT monitor core usage. 7b2d1c1b

The rationale behind this new facility is that the monitor symbol table uses 12k of memory that would normally be available for user program execution. This new feature allows the system operator to select the optimal use of memory given system load,reliability and use. 7b2d1c1c

System Startup Procedure 7b2d1d

We have modified the system such that if CHECKDSK does not run successfully, then nothing else, e.g. AUTO-STARTUP jobs, is allowed to run (except for the operator's console and one special dial-up line) until the disc has been fixed and CHECKDSK has been run successfully. 7b2d1d1

If CHECKDSK does not run successfully, then a message is broadcast to all currently connected users telling them that the disc needs fixing. 7b2d1d1a

We allow a dial-up line access in this case, so that a system programmer can fix the disc from home if neccessary. 7b2d1d1b

We made this modification with the primitive inter-job communication described below. 7b2d1d1c

Auto-start-up jobs 7b2d1e

We have changed the manner in which auto-start-up jobs get started so that they now run under the EXEC rather than under the MINI-EXEC 7b2d1e1

COMPILING AND LOADING OF A NEW MONITOR 7b2d2

We no longer add code to existing files when we get new monitor releases. Instead we have defined additional files that are assembled with each group of files and, where possible. WE have made our additions in these new files with JRSTs and CALLs to the new code. 7b2d2a

We have also broken the MON assembly into swappable and resident code similar to the SWPMON assembly. 7b2d2a1

Thus we can add code that is logically related to code in the MON assembly but not resident. 7b2d2a1a

We have made several changes in the compile-and-load sequence 7b2d2b

These changes give us more information at each step in putting together a new monitor. 7b2d2b1

We have changed both the FAIL assembly (ASSFIL) and the MACRO assemblies (MACALL) so that, where possible, we are

notified when the swappable code overlaps the resident code at compile time rather than at load time. 7b2d2b1a

When an overlap does occur, we are told about it immediately.

In addition we are told what has to be changed in order to get rid of the overlap.

To get notice of overlaps we had to break out code from PARAMS, FPARAMS, and some other routines and localize it in one routine.

The MON assembly now tells us the lower bound for loading MFLIN, etc.

In addition the driver file MACALL, after the assemblies are done, executes the subsystem TECO and types out the current load address of MFLIN

This number can then be immediately compared with the output from the MON assembly and checked for validity. We save going through a load only to discover that overlaps do occur.

We have changed the loading sequence as follows: 7b2d2b1b

We have removed the bounds checking that used to be done by going into DDT.

A new program was written that is assembled with and called by POSTLD.

This program does the checking that used to be done in DDT plus some additional checking.

In addition it outputs this information in a nicely formatted way that can be kept as current documentation for this version of the monitor.

This program also gives us the current values of certain critical cells.

We have added another program that is also assembled with POSTLD.

POSTLD calls this program.

This program types out the file names and current versions of the source files that went into generating this version of the monitor.

This information also provides useful documentation on the current monitor.

In addition to typing out current file names and version numbers it sets up cells in the monitor which contain the current version number of each of the files.

Thus we can go into MDDT and determine which source files were used to generate this monitor.

PRIMITIVE INTER-JOB COMMUNICATION 7b2e

We have implemented a very primitive inter-job communication facility. 7b2e1

It involves a system-wide cell with each bit independent of the other bits. 7b2e1a

Each bit is directly settable, resettable, and testable. 7b2e1a1

A process must know the password for any bit to set. reset, or test it. 7b2e1a2

ADVISE 7b2f

We have implemented an advise facility similar to that of the 940. 7b2f1

Its implementation is similar to the implementation of links, except that lines are checked to see if they are input linked at the time characters are put into the big buffer. 7b2f1a

If lines are input linked, then characters are placed into the big buffer with the line number of the advisee. 7b2f1a1

HANDLING OUR DISPLAYS 7b2g

(see also RFC 190 (7135,) and RFC 191 (7136,)) 7b2g1

we have made many changes to the teletype routines to accommodate our displays. 7b2g2

Basically, we defined an escape sequence: 7b2g2a

This escape sequence declares that the following n (where n is part of the escape sequence) characters are to be interpreted differently from normal TTY input. 7b2g2a1

There exists a mapping from the special sequence to normal TTY input. 7b2g2a1a

When our displays are in TTY mode (as opposed to display mode) this mapping applies.

If we are in display mode, then the characters of the special sequence include: what keyboard character was struck, what combination of the mouse buttons and keyset buttons were struck, what the current position of the mouse is, and, optionally, the time of the character input. 7b2g2a1b

This escape sequence enables us to support other types of displays (including IMLACS over the NET) with no change to either NLS or TENEX as long as these "remote" displays input the proper escape sequence. 7b2g2a1c

We have made the necessary changes to the rest of TENEX to accommadate this sequence, e.g. STI, and added additional jsies to be able to define what type of terminal (TTY, local display, remote display) is associated with each line. 7b2g2b

FAST TERMINAL HANDLING 7b2h

We have added a jsys to say that padding (sending additional rubouts) is required for this terminal when a CR or LF is output. 7b2h1

This means that the user will not lose the characters at the left margin on fast terminals. 7b2h1a

(We understand that version 1.29 takes care of padding. We will get rid of any inconsistencies in our code.) 7b2h2

SCHEDULER CHANGES 7b2i

We have changed several scheduler parameters to get the kind of response we want. 7b2il

Primarily, we have tuned the system to give very good service to highly interactive processes and very poor service to more computebound processes. We toned it by adjusting QBASE, TBASE, and TFACTR. 7b2ila

We have also changed our working set parameters in an attempt to reduce I/O wait time by getting more processes in the balance set. 7b2ilb

In addition, we have our own version of NEWST, which, together with other code, gives preferential treatment to display terminals. This is a departure from the BB&N idea of 1/N service to all users. We give our display users a larger share of the machine. 7b2ilc

Quite a bit of code has been added at ARC to gather statistics. 7b2i2

This measurement code is part of a subsystem (SUPERWATCH) written at ARC to get a profile of the system performance at any time. 7b2i2a

Documentation on this system is available (see below). 7b2i2a1

In addition to finding how the CPU's time is spent, what important scheduler variables are, and how our disc and drums are behaving, we can sample the program counter and/or the contents of memory. 7b2i2b

The program counter (PC) sampler is very valuable. The PC is sampled when the two clocks are synchronized every 50ms, in the clock interrupt. Either user or system mode is sampled. A specified subsystem may be sampled in user mode. The information is collected as a count of samples within specified ranges, with one count for out-of-range in each direction. The ranges are specified as a lower bound and word count per range. The word count is rounded to a power of two, so that a SUB, LSH, AOS sequence does the job. 7b2i2b1

The sampling of memory (user pages) is done by a process clock on 500ms intervals. It gives us a profile of memory use. 7b2i2b2

Pages are categorized as private unmodified, private

modified, shared but not referenced, shared and referenced by one process, shared and referenced by more than one process. 7b2i2b2a

Based on the information gathered by the measurement code, the scheduler generates a number we call the response index. It is an exponential average of the length of time processes on queue zero wait on the go list before being brought into the balance set. 7b2i2c

It indicates the kind of service being provided to interactive users. 7b2i2c1

If the response index goes over a threshold, the EXEC prevents new logins. 7b2i2c2

Since our scheduler has been biased against computebound jobs, we added a JSYS to set some scheduler parameters (TBASE, TFACTR). This allows us to dynamically alter the scheduling characterists of our system. 7b2i3

A special subsystem allows the operator to set the parameters to "compile time" or "normal". 7b2i3a

Compilations during regular hours do not disrupt service to interactive users (and in general, don't get done). Interactive service during compile time is poor, and compilations get done quickly. 7b2i3b

MISCELLANEOUS CHANGES 7b2j

GTJFN 7b2j1

If a version number of -4 is specified to GTJFN then the following happens: 7b2j1a

If the file exists then the user is returned a JFN for the highest version number of the file. 7b2j1a1

If the file does not exist then a file is created for the user and he is returned a JFN for this new file. 7b2j1a2

DELNF 7b2j2

This is a new jsys that we have added. 7b2j2a

It will delete all but the n (where n is a passed parameter) highest versions of a file. 7b2j2a1

Superwatch. 7b3

Abstract 7b3a

superwatch is an information gathering and formatting program designed to help find out what is going on within our TENEX timesharing system. 7b3a1

It is designed to put a very small load on the system while collecting information from it, so that it will not alter the operation of the system significantly. 7b3a2

Introduction 7b3b

The system monitoring is done in several steps: 7b3b1

Information is collected within the timesharing monitor in a crude form, usually as meters. A meter is a counter that is continually incremented, and represents a count of events or the sum of quantities. 7b3b1a

The difference between two meter readings, and the time interval between the readings, can be used to compute an average rate over the interval. 7b3b1a1

A user mode program collects the crude data from the system at specified intervals. This information is written directly on a file. This process must put very little load on the system. 7b3b1b

At the end of the collection period, the file is printed in whatever form the user desires. 7b3b1c

Monitor meters 7b3c

The TENEX system, as it came to us, contained several meters, but we found them inadequate in answering our questions about the system. 7b3c1

We added several types of information collection to the system. 7b3c2

We added meters to the monitor, especially with respect to how time was spent in scheduling and other system overhead functions. 7b3c2a

We added sampling code to a clock interrupt routine to measure several things, especially with respect to the balance set and memory utilization. The sampling is done at a 50ms. rate. 7b3c2b

Perhaps the singly most useful thing is the PC sampler, which runs off the same clock interrupt. Given ranges of program addresses, the PC sampler counts the number of times the program counter was within each range at the clock interrupt. From a large number of such counts one can infer the percentage of time spent executing in each range. The PC sampler can be focused on a specified subsystem or the system itself (system mode execution). 7b3c2c

A device called a fault record works in a manner similar to the PC sampler and records page faults. It records either fault location or fault address for a specified subsystem. The user gets a picture of where page faults occur in the program in question. It is generally used to refine program organization. 7b3c2d

The User Program - Superwatch Subsystem 7b3d

The subsystem has commands for collecting crude data from the monitor and writing it on a file, and for reading such files and formatting the output in a variety of ways. 7b3d1

The collection can be done in one of two modes. One collects a smaller amount of data, runs faster, and uses less file space. The other must collect data from the monitor at a slower rate since it takes more time. 7b3d1a

The primary parameter specifed by the user is the interval between collections or samples. The program simply dismisses itself for the specified interval between samples. 7b3d1b

The collection code has been written so that virtually no information is lost if the system crashes, or if the program is terminated by the user. 7b3d1c

After collection is finished, the statistics are obtained by formatting the file. Printouts for an entire file, or just a portion (given two times of day) can be obtained with a variety of print commands. 7b3d2

The objects of the printouts are parameters. The set of parameters is a superset of the set of meters or items collected from the monitor. 7b3d2a

Many parameters are functions of several meters. The value of some parameters are computed by a procedure which has available to it all data collected from the monitor at the time interval in question. 7b3d2a1

Generally, the user specifies a set of parameters he wishes to see. 7b3d2a2

The simplest format is a list of the values of each requested parameter at each interval. An average over the entire test is included. 7b3d2b

Fault record and PC results are printed in table form giving address ranges, counts and percentages for each range. 7b3d2c

For a specified parameter, a line printer histogram can be printed, for either the distribution of that parameter (distribution of the values at each interval), or the parameter's value as a function of time. 7b3d2d

A special command allows a real time display (histograms) of specified parameters in real time. This is actually a collection command rather than a print command. Other collection commands allow the user to request a real time printout. The formatting is done at the time of collection, and the user can see the results immediately. 7b3d2e

Typical use 7b3e

We usually use the subsystem in one of several ways: 7b3e1

We often run it with a collection interval of about 1 to 5 seconds for 10 minutes to an hour during peak loads to study performance. 7b3e1a

Another mode is to run it all day with a collection interval of 15 minutes. This gives a profile of the system usage, type of load, and overall performance for the entire day. A job which runs the subsystem in this mode is automatically started up when the time sharing system is started. 7b3e1b

The slow type sampling with a 15 min. interval is also used to collect PC and Fault record statistics. The sampling is generally done over a period of about 3 to 5 hours. 7b3e1c

The real time display mode is useful for finding out what is happening when the system is behaving strangely. 7b3e1d

There are several parameters which we have found to be very useful: 7b3e2

It is essential to know where the CPU time is going. We deal in terms of percent of real CPU time: 7b3e2a

- idle time 7b3e2a1
- time spent running user programs 7b3e2a2
- time scheduling 7b3e2a3
- time spent waiting on drum and/or disc 7b3e2a4
- time in system overhead (e.g. network, garbage collection, etc.) 7b3e2a5

Disc and Drum behavior and usage: 7b3e2b

- percent of time busy 7b3e2b1
- queue lengths 7b3e2b2
- time to transfer a page, including queue wait time 7b3e2b3
- number of reads, writes 7b3e2b4

Memory utilization: 7b3e2c

- number of jobs holding space in memory 7b3e2c1
- amount of memory reserved for above jobs 7b3e2c2
- actual number of pages held by above jobs 7b3e2c3
- number of free pages 7b3e2c4
- number of pages retained due to sharing 7b3e2c5

usage by subsystem 7b3e2d

- percent of real timg used 7b3e2d1

compute time between page faults 7b3e2d2

There are many other parameters (over 100). They nearly all are of value just to know that various aspects of the system are functioning well. 7b3e3

A Few Discoveries 7b3f

several times our Bryant Disc has malfunctioned in such a way that it took the maximum length of time to do a seek. As a result, disc transfers were very slow (about 260ms. per page) and the system response very poor. It was not apparent that the disc was the culprit since no errors were being reported. But a statistical printout showed the long disc page times, as well as a long disc queue length, and excessive I/O wait and low utilization because of the disc. 7b3f1

we also discovered a performance problem in the time sharing system. When many jobs were sharing the same subsystem, the system was over-reserving memory for those jobs. We presented the problem to BB&N, and the next version of TENEX from BB&N had a modified memory management package in it which handled shared pages in a more satisfactory way. 7b3f2

The PC sampler has uncovered two expensive parts of the scheduler which may have been corrected in the newest release from BB&N (TENEX 1.29 which we have not used yet). Also, the PC sampler has been a guide for reorganizing the code in NLS, in order to group frequently used code to reduce the working set size. 7b3f3

we keep track of the overhead time spent handling the Network. It is in the range of 0.5% to 1% of the real CPU time per NET user to maintain it. 7b3f4

one of our subsystems makes particularly heavy use of the disc (BSYS). when running on the Bryant disc, we found that system performance was very poor when BSYS was running and using the disc heavily. A statistical printout showed that it was due to very high I/O wait time because of a long disc queue. This was a factor (in addition to reliability) for getting the disc packs. When running BSYS with the disc pack system, the disc use is increased, but the I/O wait time is not significantly increased. The page transfer time on the Bryant disc is about 160ms., and on the packs it is about 35ms. 7b3f5

occasionally, we have problems with one 16K memory box. The

standard procedure is to run without it, with 16K less memory for swapping space. The result is a very clear degradation in service, with more time spent in I/O wait and with fewer jobs in memory at one time. Also, we occasionally run with the system DDT (debugging system) and symbols resident in memory. This reduces user swapping space by about 10K, and the result is evident in a statistical printout. This prompted us to think that we would gain in system performance by increasing the amount of memory. Another 32K will be delivered soon, and we shall see just how much it increases our performance. 7b3f6

Generally, an information gathering system like ours is valuable: 7b3f7

To verify that the system is working as designed. 7b3f7a

To identify the cause of poor service at the time it is happening (e.g. a bug, hardware malfunction, or just overloading). 7b3f7b

To identify the "weak link" in the system configuration (drum, disc, memory or CPU capacity). 7b3f7c

To evaluate changes in the system or hardware configuration. 7b3f7d

REFERENCES IV - Computer Facility 7c

(7135,) Peter Deutsch (PARC). NWG/RFC 190, DEC PDP-10 -- IMLAC Communication System. ARPA Network Information Center, Stanford Research Institute, Menlo Park, California 94025. 13 July 1971. 15p. 7c1

(7136,) Charles Irby (SRI-ARC). NWG/RFC 191, Graphics Implementation and Conceptualization at ARC. ARPA Network Information Center, Stanford Research Institute, Menlo Park, California 94025. 13 July 1971. 4p. 7c2

(8277,) D. C. Engelbart (SRI-ARC). Network Information Center and Computer Augmented Team Interaction, Interim Technical Report. Augmentation Research Center, Stanford Research Institute, Menlo Park, California 94025. Rome Air Development Center, ARPA. RADC-TR-71-175, AD 737 131. 30 June 1971. 104p. 7c3

SRI-ARC 8 JUNE 1972 13041
Plans
Goals

PLANS

by James C Norton, Richard W Watson#@
and Douglas C Engelbart 8

GOALS 8a

ARC plans to resolve a set of interdependent goals by conducting research and providing service under a new "Base-Project" contract that concentrates primarily upon the goals of: 8a1

Advancing the techniques available to ARC and Network system builders and users for augmenting the development Future Plans of computer-based information systems. 8a1a

Making the Network Information Center into both 8a1b

(1) an increasingly useful service to the Network Community and 8a1b1

(2) an important part of the Network Experiment (in its distributed, collaborative operations and in its Network-utility role). 8a1b2

Moving useful augmentation techniques and services out into the ARPA-Network Community. 8a1c

In the discussion that follows, and in our proposal to RADC/ARPA (7404,), we outline the types of activity that seem to us best to meet these goals. 8a2

SERVICE TO USERS 8b

A central point of our proposed approach is that we need to become prepared to negotiate and provide an extensive amount and range of services to distributed users. Our position stems from the following reasoning: 8b1

Our planned NIC services involve a steadily expanding set of explicit "reference and dialog support" services (see -- 7406,). This is considered by us to be the central commitment of a "Network Information Center." We plan to be ready to expand the operational capacity of these services as needs and possibilities emerge. 8b1a

Aside from these NIC-explicit services, there are other services that our general set of tools and methods can provide and that are of interest to other parties. Over the years that the Network has

been evolving, there have been many discussions about the potential value ARC's tools might have for different Network individuals and groups. Recently there has been a distinct increase in interest and expectation in this regard. 8b1a1

In general, we enjoy this show of interest in our products, and in particular we want very much to collaborate with and support some of this experimentation (as in the goal set cited above). 8b1a2

However, it is quite obvious to us that significant value will not be obtained from extra-NIC experiments with our computer services, or from interaction with our staff, unless these be done in a nondissipative way, with individuals or groups 8b1b

(a) Whom we can adequately support with computer and personnel resources, and 8b1b1

(b) That show promise of following through, by being able to acquire adequate resources and being able to integrate our services significantly into the work that they will be doing. 8b1b2

Furthermore, it is also obvious to us that there will be considerably more payoff (to our and ARPA's goals) from the external use of our finite resources, if these are individuals or groups interested in bootstrapping -- that is those who 8b1c

(c) Will pursue activities that either add to the techniques and capabilities subsequently available to other participants, or who will help other people learn about and obtain this kind of service. 8b1c1

On another tack, if the concept of a distributed community making use of "network utilities" is to materialize, then certainly there must evolve a body of techniques and conventions involving 8b1d

(a) Service Delivery--where these utilities can deliver responsive, interactive transactions, over a complex repertoire of service functions, with both a high degree of reliablity and a high degree of availability, and 8b1d1

(b) Service Marketing--where a customer can negotiate with a utility for the quantity and type of service that suits his needs and where there is a negotiation environment at service-transaction time that enables the customer to get the service when he needs it, but with a resource-utilization framework that is balanced between efficiency and demand capacity. 8b1d2

Therefore, we plan to concentrate our efforts within a four-pronged project wherein coordinated advances can be made in: 8b2

(1) Developing service functions that will be of maximal value in our above-mentioned goal structure, 8b2a

(2) Developing the knownow and capability for delivering significantly useful service to the Network, as a utility, 8b2b

(3) Developing the knownow and capability for marketing a utility service to the Network, 8b2c

and wherein we become ever better at 8b3

(4) Operating a utility service. 8b3a

Depending on funding availability and other arrangements to be negotiated we may find ways to provide additional service capacity through placement of the computer-based portion of our augmentation system on a computer or computers operated for us a commercial timesharing utility. 8b3a1

BASIC PROJECT WORK 8c

We are planning that under our new base contract, ARC's "utility" would initially serve two, bulk-commodity customers--ARC workers and NIC customers. 8c1

Until we learn how to market and deliver service better, we would rather concentrate heavily upon developing our marketing and delivery capabilities, as contrasted with expending a large amount of energy in trying to meet the beyond-basic-NIC services that might be wanted by "customers." 8c1a

And as we learn how to deliver and market different types and quantities of service, we feel that there will be a logical progression of service types and of customer types to be effectively and beneficially promoted and served in our growing "utility market." 8c1b

We outline below what seems to be a natural succession of "service systems" that might be thus marketed, and we would propose concentrating our service-function development efforts on getting prototypes of these service systems shaken down within ARC's internal domain in readiness for marketing them when the time is right. 8c1b1

we will hope to develop a market for our services that leans strongly toward-B customers interested in bootstrapping. 8c1c

If we make unexpected progress in developing delivery and marketing capability, and if it appears that additional ARPA funding could profitably be allocated for "buying" more service for some types of utility customers, we assume that the utility service provided under the contract would be extended beyond that initially negotiated. 8c2

Basically, we expect that the Base Project will count on putting a significant and constant effort into continuous development of delivery/marketing techniques and principles and that any expansion of ARC's service-delivery capacity be supported by means of explicit additional negotiations with customers (and perhaps with the customers' sponsors). 8c2a

Our Base Project work will focus on: 8c3

(1) Developing Service Functions for: 8c3a

(a) External Users (via the Network) 8c3a1

NIC reference and dialog support functions (discussed further in -- 7406,) 8c3a1a

Our planned major points of emphasis are as follows: 8c3a1a1

Continue to work with Network Working Groups, particularly in those areas vital to the NIC such as graphics, file transfer, distributed data management, and accounting. 8c3a1a1a

Expand our ability to provide basic reference and dialog support for the increasing numbers of network users and groups who will be coming on the NET. 8c3a1a1b

Reorganize our hardware and software system to enable smooth expansion as the need arises. 8c3a1a1c

Get our resource accounting of both people and machine resources in shape so as to be able to know what each operation and service is costing. 8c3a1a1d

As new dialog support functions are developed and tested on the research side of the house, move them into operation in the NIC. 8c3a1a1e

Provide improved querying capabilities for the online reference files such as the: 8c3a1a1f

NIC catalog
Network Resource Notebook
The Current Network Protocols
Records of site status
Documentation of site facilities and services
Networkwide and personal files of people interested in various research topics

Possibly provide a facility to ask questions for online updating of site status or other files that are changing over a short period of time. 8c3a1a1g

Continue to improve making information available by preparing weekly notices of new additions to the NIC collection. 8c3a1a1h

Prepare specialized bibliographies for subjects of wide interest. 8c3a1a1i

The above services as well as evolving the NIC collection require considerable effort to:

Monitor current literature to select, collect, abstract, and catalog

Design and program to produce such listings from catalog input items

Prepare and distribute

Devise improved ways to handle hardcopy at sites: 8c3a1a1j

As the number of users grows and the number of available services increases, the size of the collections at local sites will increase.

Allow individuals and groups the capabilities of NIC to create and manage their own private collections of information with catalogs and capabilities for entering and proofing items and querying the catalogs. 8c3a1a1k

This item requires basic bibliographic tools beyond those used for producing the standard NIC catalog.

It places more requirements for training and close liaison with users.

Since these users will have online items, possibly scattered in files on other hosts, it would be desirable to provide ways for retrieving them through their catalogs in NIC.

Learning to use all the varied systems on the network is not going to be possible by sending all potential users to each remote site for training or by always bringing instructors to the user's site. 8c3a1a11

(b) Internal Users (evolving toward external use), such as: 8c3a2

Prototype Dialog Support System
(discussed further in -- 7407,) 8c3a2a

Our DSS development will be coordinated with our other developments toward serving teams of people involved in developing complex computer-based systems. Hence we will concentrate upon making a prototype DSS that really supports the developers and users of the systems that ARC is developing and operating--such as: NLS, NIC, DSS, BRS and, DPCS. 8c3a2a1

The ARC Handbook is the prototype "super document" (see -- 5220,5b) that our collaborative dialog will concentrate upon for ARC's internal, prototype development of DSS. 8c3a2a1a

As features of DSS are seen to be useful to the NIC system of services, they will be so provided. This will provide us with early experience in the use of DSS features among a larger, distributed community. 8c3a2a1b

For instance, we expect to use improved link and/or advise features in simultaneous online conference dialog and other working collaboration when and where consistent within ARC and NIC goals.

It is assumed that there may be special DSS features required for this distributed-community service; we expect to do this sort of work within the "functional-development" part of our activity. Some discussion of the "distributed-dialog" features is to

be found in a 7 Dec 1969 memo.
(see -- 5220,5e)

Examples of functions under development or being considered are: 8c3a2a1c

Sets--the ability to find those items in the dialog universe relevant to one's interest and view them in many ways.

Backlinks--to find out which other items are referencing each item.

Ability to build sub catalogs of dialogs--related to sets.

Dialog with files distributed in many hosts throughout the network--There are many problems that would have to be solved such as assuring that files did not get deleted and keeping track of where things are in our catalog, to help reduce the load on NIC.

Action items--ways to enter a dialog item requiring action by a certain date and having the system remind the sender to follow up or check to see whether the receiver responded.

New Journal entry techniques--making the process much easier for users, including pre-specification in NLS files of entry details.

Prototype Documentation Production and Control System (DPCS)
(discussed further in -- 7408,) 8c3a2b

We plan to further develop within ARC a separate place, terminal configuration and staff -- for a DPCS system expressly to support production and control of information-systems' documentation -- where the support work for developing and controlling ARC's documentation will all be done. 8c3a2b1

If we need more throughput to shake down the system, and/or if through NIC's activity or through special arrangements with Network groups there is reasonable sense to do so, we would consider our supporting of other-group (Network) documentation and control. This

activity would serve as a test bed for the successive stages of DPC system developments we want to go through. 8c3a2b1a

In support of this approach, we would like to provide ourselves locally with a hardcopy printout system capable of making good-quality mixed-text/graphic drafts. 8c3a2b2

We may coordinate this system with parallel use of a high quality COM system for final production of documents and microfiche. We expect to use a commercial service bureau for this initially, but want to have our local facility be capable of producing completely accurate representations of the final output. 8c3a2b2a

The earliest form of a DPCS would have the authors working offline to a large extent, using Deferred Execution (DEX) processes. 8c3a2b3

The features provided by DEX will allow clerical people to follow complex mark-up notations made by authors as drafts are developed. 8c3a2b3a

Spooled-input typewriters, using magnetic tape recording equipment would be used. 8c3a2b3b

Subsequent evolution of a DPCS would be towards providing features such as: automatic concordance-type indexing, cross-reference control, glossary control and production, and towards extended representations, new forms of portrayal for use in documenting complex systems, an extended facility for composing and modifying the exotic representations, and high-quality font/formatting. 8c3a2b4

Prototype Software-Engineering Augmentation System (discussed further in (7409,) and -- 7411,) 8c3a2c

New or changed features being considered are: 8c3a2c1

Source level debugging and incremental compilation 8c3a2c1a

A primitive system is being implemented now with a more advanced version to come following MPL implementation.

Documentation aids 8c3a2c1b

Documentation for different levels -- user guide level, system architecture level, etc.

Automatic documentation -- reformatter programs to make embedded documentation more clear and in summary form.

Coding aids 8c3a2c1c

Possibly produce parsers automatically from use-level documentation of commands -- that would require fairly strict rules for documentation.

Use of back-linking for cross-reference and annotative documentation. 8c3a2c1d

Develop cross-reference facility for modules (and possibly inter-modular if back-links are not sufficient).

Use set facilities for viewing and working on code at different levels. 8c3a2c1e

For example, a user might construct a set that contains all procedures that are pertinent to a particular DNLS command, or a set of all procedures that do command parsing for DEX and so forth.

Further develop our compiler system 8c3a2c1f

Allow easy generation of interpreters as well as compilers.

Modularize compilers.

Possibly generalize trees to netwoKS IN TREE META.

Develop program verification capabilities 8c3a2c1g

Development of heuristic programs to determine that programs do what the writer has indicated they do. Allow statement of programmer asumptions at various points in a program. The verification facilities might test those assumptions and specify if they can be false.

SRI-ARC 8 JUNE 1972 13041
Plans
Basic project Work

Develop trace facility in the Baseline program 8c3a2c1h

To help immprove estimating and give a history of our activity.

Develop catalogs and indices for system files 8c3a2c1i

Index procedures by function.

Once a remote site has established a DNLS station that can work with our system through the Network, it would be directly feasible for software engineers, working on other computers with other languages than ours, to use our DNLS system to considerable advantage as a workshop in which to compose, modify, and study their (integrated) source code and documentation, and to participate in computer-aided, collaborative dialog over this material. 8c3a2c2

With straightforward utilization of our compiler-compiler techniques operable through DNLS, they can easily build special-purpose languages that match to other computers, to other purposes, at binary or assembly-language levels. 8c3a2c2a

We hope to encourage some experimentation in this direction, and intend to round out the prototypical set of conventions, aids, principles, etc. within our application areas that will make such application relatively direct. The extent of such experimentation will of course be limited to what we can manage to support, both with computer service and with people interaction. 8c3a2c2b

More Advanced Use of the Software Tools at ARC 8c3a2c3

We have described above how the software engineer at another location might use NLS for writing his programs. 8c3a2c3a

It will also be possible for the remote programmer to use other software augmentation tools developed here. 8c3a2c3b

For instance, the TREE META compiler writing system could be modified to produce code for another machine (this was in fact done as part of the transfer of NLS from the XDS-940 to the PDP-10). It could then be used to develop experimental compilers that would run on a PDP-10 (or through further modifications and

bootstrapping, on another machine) and produce files that could be sent over the Net for loading.

The feasibility of such an undertaking will be greatly increased with the development of the Modular Programming System described in (7411,). 8c3a2c3c

The compiler-compiler will be composed of modules, so that the code production can be more easily replaced without requiring a detailed understanding of large sections of a complex program.

When it is operational, the Modular Programming System itself will be a very powerful tool and of interest to other programmers. 8c3a2c3d

In addition, it will open up new ways for the remote programmer to access and use the other tools at ARC. 8c3a2c3e

It will become possible for the programmer to create a personal version of NLS by the replacement and addition of modules so as to better match his needs.

Prototype System-Developer's Handbook System 8c3a2d

We will design the next stage Handbook, including the specification of content categories together with techniques and procedures for maintenance of the Handbook. 8c3a2d1

We plan to implement this design during the next contract period. 8c3a2d1a

Indices and Tables of Contents for the Handbook are planned to be designed and implemented. 8c3a2d1b

We expect to complete the collection of the basic existing Handbook-relevant documents that already exist, both in hardcopy and online files. We will also add new Handbook-relevant documents as they are produced, retiring obsoleted documents as appropriate. 8c3a2d2

We expect the Handbook system to aid in stimulating the production of documents that are needed, but missing from our information base. 8c3a2d2a

Prototype Baseline Record System
(discussed further in -- 7410,) 8c3a2e

We plan to improve the data collection procedures and storage mechanisms of the present Baseline Record System. 8c3a2e1

Our present Baseline data storage techniques will probably be changed to use a more generalized system common to the Baseline system, Catalog system, and other ARC data handling systems. 8c3a2e1a

ARC users will be more effectively oriented toward the need for and trained in a more organized task definition and selection process. 8c3a2e2

Better views of the Baseline Record will be produced for use by ARC, with more useful user-created view capabilities provided. 8c3a2e3

Hardcopy and online Baseline Records will be more complete and made a part of the daily working life of ARC researchers. 8c3a2e4

We plan to develop better methods for keeping the Record up-to-date, both online and in hardcopy. 8c3a2e5

The ARC resource accounting system, as it develops, will be integrated with and used by the Baseline Record System. 8c3a2e6

(2) Developing Service-Delivery Principles and Practices for: 8c3b

(a) Computer services, including considerations such as: 8c3b1
Remote DNLS 8c3b1a
Remote Hardcopy Delivery 8c3b1b
Reliability 8c3b1c
Resource allocation, accounting, billing 8c3b1d
The questions of scale, efficiency, reliability 8c3b1e
Service-capacity expansion plan 8c3b1f

(b) NIC-service (information, people help) 8c3b2

(c) Transcription services 8c3b3

(d) Documentation services (as operational prototype of DPCS, within ARC). 8c3b4

(3) Developing Service-Marketing Principles and Practices: 8c3c

(a) Learning how to negotiate with prospective customers for delivering various kinds of service to them, including questions such as: 8c3c1

On what basis are the agreements made? 8c3c1a

How is financing accomplished? 8c3c1b

How is the accounting performed? 8c3c1c

How are the scheduling and billing of service delivery accomplished? 8c3c1d

How are conflicts resolved (market conventions, arbitration)? 8c3c1e

What guarantees can practically be made regarding, accessibility, reliability, documentation accuracy and completeness, and the like? 8c3c1f

How are user training and helping provided? 8c3c1g

(4) Providing Operational Marketing and Delivering of Services: 8c3d

Developing the framework, as the marketing and delivery systems begin to take shape, in which the current service resources are marketed within the ARC and NIC customer market. 8c3d1

Studying the possibilities of evolving the various "prototype" services into marketable items, negotiating the resources for this, extending our service market--all in an orderly process involving a number of multiparty agreements. 8c3d2

REFERENCES 8d

(5220,) Douglas C. Engelbart (SRI-ARC). ARPA Memo #2. Augmentation Research Center, Stanford Research Institute, Menlo Park, California 94025. 7 December 1969. 15p. 8d1

(7404,) Proposal for Research [to RADC/ARPA ISU 71-94]. Network Information Center and Augmentation System Development. SRI No. ISU 71-94. Augmentation Research Center, Stanford Research Institute, Menlo Park, California 94025. 29 July 1971. Separately paged. (Entered in SRI-ARC Journal 18 October 1971.) 8d2

(7406,) Appendix B: NIC Development - Function and Operational Delivery. In: Proposal for Research /to RADC/ARPA ISU 71-94/. Network Information Center and Augmentation System Development, (7404,). SRI No. ISU 71-94. Augmentation Research Center, Stanford Research Institute, Menlo Park, California 94025. 29 July 1971. 11p. (Entered in SRI-ARC Journal 15 October 1971.) 8d3

(7407,) Appendix C: Dialog Support System. In: Proposal for Research /to RADC/ARPA ISU 71-94/. Network Information Center and Augmentation System Development, (7404,). SRI No. ISU 71-94. Augmentation Research Center, Stanford Research Institute, Menlo Park, California 94025. 29 July 1971. 7p. (Entered in SRI-ARC Journal 15 October 1971.) 8d4

(7408,) Appendix D: Documentation Production and Control System. In: Proposal for Research /to RADC/ARPA ISU 71-94/. Network Information Center and Augmentation System Development, (7404,). SRI No. ISU 71-94. Augmentation Research Center, Stanford Research Institute, Menlo Park, California 94025. 29 July 1971. 3p. (Entered in SRI-ARC Journal October 1971, but not presently online.) 8d5

(7409,) Appendix E: Software-Engineering Augmentation System. In: Proposal for Research /to RADC/ARPA ISU 71-94/. Network Information Center and Augmentation System Development, (7404,). SRI No. ISU 71-94. Augmentation Research Center, Stanford Research Institute, Menlo Park, California 94025. 29 July 1971. 4p. (Entered in SRI-ARC Journal 18 October 1971.) 8d6

(7410,) Appendix F: Descriptive Notes About PBMS, a Project Baseline Management System. In: Proposal for Research /to RADC/ARPA ISU 71-94/. Network Information Center and Augmentation System Development, (7404,). SRI No. ISU 71-94. Augmentation Research Center, Stanford Research Institute, Menlo Park, California 94025. 29 July 1971. 3p. (Entered in SRI-ARC Journal 18 October 1971.) 8d7

(7411,) Appendix G: Collaborative System-Evolution System. In: Proposal for Research /to RADC/ARPA ISU 71-94/. Network Information Center and Augmentation System Development, (7404,). SRI No. ISU 71-94. Augmentation Research Center, Stanford Research Institute, Menlo Park, California 94025. 29 July 1971. 5p. 8d8

GLOSSARY 9

APR -- Acronym for the Arithmetic Processor of the PDP-10. 9a

ARC -- Acronym for Augmentation Research Center. 9b

ARPA -- See DARPA 9c

Arithmetic Processor -- Xhe central processing unit of the PDP-10. 9d

Augmentation -- In this report, extension, improvement, or amplification of human intellectual and organizational capabilities by means of close interaction with computer aids and by use of special procedural and organization techniques designed to support and exploit this interaction. 9e

BB&N -- Bolt Beranek and Newman. A commercial research and development organization under contract to ARPA for services to the ARPA Network, and under other contracts that lead to frequent interaction with ARC. 9f

BRS -- Acronym for Baseline Record System 9g

Baseline Record System -- Part of a developing management system used at the center. It records tasks and people assigned to tasks, and allows retrieval of information about tasks by people or people by tasks. 9h

Bootstrapping -- A name for the research strategy of the ARC. By "bootstrapping" we mean taking advantage of the feedback in recursive development of systems. That is, we try to test ways of augmenting intelligence by their usefulness in developing new systems to augment intelligence, through the use of the new system features by (mainly) the developers of the system. 9i

Branch -- In the NLS hierarchy of statements, a statement and all substatements that depend on it. 9j

Bug -- The cursor visible on an NLS Display which is controlled by the hand-held mouse and which may serve as an address in NLS commands. 9k

Center -- The same as ARC. 9l

Compiler -- A computer language that is used to translate from one set of symbols to another, particularly to machine language. 9m

Console -- As used here, specifically a user's control console for the

ARC's Online System (NLS). The consoles presently in use at ARC consist of a display screen, a keyboard, a "mouse", and a "keyset". 9n

Current Statement -- In NLS, normally the last statement modified, executed, or reproduced by the user and, hence, the statement that starts the sequence of the sequence generator which generates the display image. Usually the statement at the top of the screen is the current statement, but content analysis or screen splitting may displace or obscure it. 9o

Current Statement Pointer -- The internal symbol fixed on the current statement by NLS. 9p

DARPA -- Acronym for the Defense Advanced Research Projects Agency of the Department of Defense. 9q

DDT -- Acronym for Dynamic Debugging Tool, a program useful for establishing at what point in another program a problem occured. 9r

DEC -- Acronym for Digital Equipment Corporation, the manufacturer of the Center's PDP-10 computer and RP02 disc memory. 9s

DSS -- Acronym for Dialog Support System 9t

Dialog Support System (DSS) -- The system of files, programs, and procedures at ARC for storing, sorting, and recovering the interchange of thoughts, plans, memos, technical documents, etc. that accompany our system development. 9u

Display Start Statement -- The same as "current statement" 9v

Executable Text -- In NLS, as it operated on the XDS-940, a program or subroutine that was written by users in characters as all or part of a statement and that can be carried out by a simple command from the user. 9w

FRAMAC -- From Framework Activity. An organized activity among members of the Center who are involved in planning to define long and short term goals. 9x

Field operations -- In programming NLS, manipulations that involve the capacity of the PDP-10's software to handle parts of words. 9y

File -- In NLS, this refers to a unified collection of information held in computer storage for use with the Online System. A file may contain text (English or program code), numerical information.

graphics, or any combination of these. Conceptually, a file corresponds roughly to a hard-copy document. 9z

Frozen Statements -- In using NLS, statements shown stationary on the display while other parts of the file are in view and viewed, composed or modified. 9a@

HLP -- Acronym for Higher Level Processes 9aa

Handbook -- A complete reference work of all systems and activity at the Center at a given time. 9ab

Higher Level Processes -- A phrase once used for what we now call User Programs. 9ac

IMLAC -- The manufacturer of a display console used experimentally with NLS. 9ad

IMP -- Acronym for Interface Message Processors. Hardware devices that code and decode messages for transmission between the computers on the ARPA Network. 9ae

Ident -- a two-to-six-letter code, given to people or groups for recognition by the Journal Ident system. 9af

Intellect -- The human competence to make, sort, exchange, and apply knowledge to decision making. 9ag

Journal -- The open ended information storage and retrieval system that forms the core of the Dialog Support System. 9an

JSYS -- lit.: "jump to system" The machine instruction used in Tenex to invoke a monitor supplied service; i.e. a jump to a subroutine. 9ai

Keyset -- A device with five keys like piano keys for entering characters into NLS at a display console. Each key controls a bit in 5-bit ASCII code. 9aj

L-10 -- The algol-like language in which our online system is written. 9ak

Level-clipping -- With reference to NLS Viewspecs, the practice of controlling how deeply into the outline structure of a file you see in any given view. 9al

LINAC -- From Line Activity. The line management structure of the

Center, a matrix of projects and functional organization, as differentiated from PODAC and FRAMAC. 9am

LINKS -- In NLS, a routine to search the disc for any statement and set viewspecs. Links may be part of file text and may be used as an address in TNLS commands without regard to what file the user has loaded. Links have the form (ddd,fff,n:x) where the field ddd comtains a TENEX directory name; the field fff contains a TENEX file name; tne filed n contains and NLS statement name or number and the filed x contains NLS viewspecs. Fields are frequently left to default in practice. 9an

List -- In the NLS hierarchy, the list of a given statement is the set of statements that are in the plex of the source of the given statement and are on the same level with it. 9ao

MPS -- Acronym for Modular Programming System -- A reorganization of NLS code into modules that may be exported separately and which pass control only through defined ports. 9ap

Markers -- A symbolic name that the user may attach to a particular character in a file. It is not displayed or printed, but is visible to routines that search for it. 9aq

Monitor -- A program which remains in memory at all times and controls the coming and going of data and other programs in the machine. 9ar

Mouse -- A round-topped, handsized device normally operated by the user's right hand when using the Online System from a display console. The mouse rolls freely on any flat surface, causing a cursor spot on the display screen to move correspondingly. 9as

NGG -- Acronym for Network Graphics Group 9at

NIC -- Acronym for Network Information Center, one of ARC's key roles in the ARPA Computer Network. The NIC is a computer-assisted reference and communication service for information pertaining to the Network. 9au

NLS -- Acronym for the ARC Online System. 9av

NWG -- Acronym for Network Working Group 9aw

Network Working Group -- A group of users of the ARPA Network organized to develop Network functions. 9ax

Online System -- This is ARC's principal and central computer-based

development in the area of computer aids to the human intellect. As presently constituted, it is a time-shared multi-console system for the composition, study, and modification of files (see definition of "file"). Many details of the system are described in the body of this report. 9ay

Output Processor -- The subsystem of the portrayal generator that processes NLS files into sequential files suited to drive devices that produce hard copy. 9az

PDP-10 -- The computer used at the Center from the winter of 1970 until the present. The asynchronous arithmetic Processor has a 1-microsecond cycle and uses 36-bit words paged in a BB&N paging box into 512-word pages. 9b@

POD -- Within PODAC, a group of about 8 employees of the Center that meets weekly for purposes of personal and organizational development. 9ba

PODAC -- Acronym for the continuing, organized Personal and Organizational Development Activities within the Center. 9bb

Plex -- In the NLS hierarchy, the set of all statements that have a common source. 9bc

Pointer -- An old name for marker. 9bd

Portrayal Generator -- The class of NLS code that creates something formatted for view by a human. 9be

Protocol -- Among users of the ARPA Computer Network, a document describing conventions for carrying out some activity over the Network. 9bf

RADC -- Acronym for Rome Air Development Center. 9bg

RFC -- Acronym for Request for Comments 9bh

Request for Comments --a series of memoranda between Network Liaison personell numbered and distributed at the Network Information Center. They are no longer restricted to requests for comments. 9bi

SRI -- Acronym for Stanford Research Institute 9bj

STID -- Acronym for statement identifier. A number unique to each statement in a file and that remains with the data regardless of document structure change. 9bK

Sequence Generator -- A routine that, when given the number that identifies a statement internally (the STID), will search through the file and find all the subsequent statements that observe the current viewspecs. 9bl

Server Telnet -- See Telnet 9bm

Sublist -- In the NLS hierarchy, the first sublist of a statement is the set of statements immediately below it, the second sublist is all statements one level below them. T the nth sublist of statement "s" is the set of statements that are in the first sublist of the statements in the (n-1)th sublist of "s". 9bn

Statement -- The basic structural unit of a file. A statement consists of an arbitrary string of text, plus graphic information. A file consists of a number of statements arranged in an explicit hierarchical structure. 9bo

Superwatch -- A group of programs that measures the loads on different pieces of hardware and on subsystems of TENEX and NLS. 9bp

TENEX -- The timesharing system that supports NLS on the PDP-10. NLS runs as a subsystem of TENEX and draws extensively on TENEX's file handling. 9bq

TNLS -- Acronym for Typewriter Online System. The system used at ARC for typewriter type terminals from early 1971 on. It differs from TODAS internally in using core NLS with adaptive routines that are called automatically when the user names his terminal in logging in, and externally in a number of additional, powerful editing commands. 9br

TODAS -- Acronym for Typewriter Oriented Documentation Aid System. The version of NLS used from typewriter-like terminals prior to 1971. 9bs

Telnet -- In the ARPA Network, the software that allows a user at one site access to a time-sharing system at another site. User Telnet is the software at the user's site; Server Telnet is the software at the remote site. 9bt

Textpointer -- In NLS, as used on the PDP-10, the fixation by NLS on a space between two characters which allows the users to be sure editing will begin with the following character. 9bu

Tree Meta -- The ARC compiler-compiler system, used to compile all the languages at ARC. 9bv

User Programs -- Processes in which the basic user features of our online systems (particularly NLS and TNLS) are used as building-blocks in the construction of programs for carrying out specific, perhaps rather complicated tasks. 9bw

User Telnet -- See Telnet 9bx

Viewspecs -- A feature of NLS whereby a user may mask part of his files, such as the hierarchical numbering, or statements below a certain outline level, in order to better view the unmasked portion. 9by

XDS -- Xerox Data Systems, manufacturer of the XDS-940 Computer, used at the Center until January of 1971. 9bz

SRI-ARC 6 JUNE 1972 13041
Appendices
Handbook Contents

APPENDICES

10

APPENDIX I CONTENTS OF ARC HANDBOOK

11

Section 1. USER DOCUMENTS

11a

Volume I NLS GENERAL CATNUM

Section 1. USER DOCUMENTS

11b

Volume II TNLS SPECIFIC CATNUM

Section 1. USER DOCUMENTS

11f

Volume VI CATNUM

Section 1. USER DOCUMENTS

11g

Volume VII CATNUM

Section 2. SYSTEM DOCUMENTS

Section 2. SYSTEM DOCUMENTS

11j

Volume III - TENEX CATNUM

GENERAL

JSYS

April 1961
Prepared for: Annual Meeting of American Documentation Institute, Berkeley, California

Augmenting Human Intellect: A Conceptual Framework 3906
October 1962
Prepared for: Director of Information Sciences Air Force Office of Scientific Research, Washington 25, D.C.

A Conceptual Framework for the Augmentation of Man's Intellect 9375
1963
Prepared for: Director of Information Sciences, Air Force Office of Scientific Reseearch, Washington 25, D.C.

Augmenting Human Intellect: Experiments, Concepts, and Possibilities 9691
March 1965
Prepared for: Directorate of Information Sciences, Air Force Office of Scientific Research, Washington 25, D.C.

Research on Computer-Augmented Information Management 9690
March 1965
Prepared for Directorate of Computers, Electronic Systems Division, Air Force Systems Command, United States Air Force, L. G. Hanscom Field, Bedford, Massachusetts

Computer-Aided Display Control 9692
July 1965
Prepared for: National Aeronautics and Space Administration, Langley Research Center, Langley Air Force Base, Virginia

Section 4. REPORTS

11n

Volume II 1967-1968 CATNUM

Display-Selection Techniques for Text Manipulation 9694
March 1967
Prepared for: IEEE Transactions on Human Factors

Prepared for: Directorate of Computers, Electronic Systems Division, Air Force Systems Command, United States Air Force

Section 4. REPORTS

Volume III 1969-1970 CATNUM 11n

Study For the Development of Human Intellect Augmentation Techniques 3562
January 1969
Prepared for: National Aeronautics and Space Administration, Langley Research Center, Mail Stop 126, Langley Station Hampton, Virginia 23365

Study For The Development of Human Intellect Augmentation Techniques 9701
18 February 1969
Prepared for: National Aeronautics and Space Administration, Langley Research Center, Langley Station, Langley, Virginia 23365, Mail Stop 126

Study For The Development of Computer Augmented Management Techniques 9703
8 March 1969
Prepared For: Rome Air Development Center, Air Force Systems Command, Griffiss Air Force Base, New York

Study For The Development Of Human Intellect Augmentation Techniques 9704
18 August 1969
Prepared for: National Aeronautics and Space Administration Langley Research Center, Langley Station, Langley, Virginia 23365, Mail Stop 126

Study For The Development of Computer Augmentation Techniques (Part One) 5140
November 21, 1969
Prepared for: Rome Air Development Center, Griffiss Air Force Base, Rome, New York 13440

Computer-Augmented Management-System Research and Development of Augmentation Facility 5139
April 1970

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

Intellectual Implications of Multi-Access Computer Networks 5255
April 1970
Prepared for: Interdisciplinary Conference on Multi-Access Computer Networks

ARPA Memo #2, Regarding Relevance of our proposed work to DOD activities or problems 5220

Section 4. REPORTS

Volume IV 1970-1972 CATNUM

Advanced Intellect-Augmentation Techniques 5140
July 1970
Prepared for: National Aeronautics and Space Administration, Langley Research Center, Langley Station, Mail Stop 126, Hampton, Virginia 23365

Network Information Center and Computer Augmented Team Interaction 8277
18 May 1970
Prepared for: Advanced Research Projects Agency Washington, D.C. and Rome Air Development Center, Griffiss Air Force Base, New York 13440

1970 ARC ACTIVITY SUMMARY 5642
5 FEB 71

QUARTERLY MANAGEMENT REPORT 1 (Covering Period 9 February 1970 through 8 May 1970)
QUARTERLY MANAGEMENT REPORT 2 (Covering Period 9 May 1970 through 8 August 1970)
QUARTERLY MANAGEMENT REPORT 3 (Covering the period 9 August 1970 through 8 November 1970)
QUARTERLY MANAGEMENT REPORT 4 (Covering the period 9 October 1970 through 8 April 1971) 6638

QUARTERLY MANAGEMENT REPORT 5 (Covering the period 9 February 1971 through 8 May 1971)
QUARTERLY MANAGEMENT REPORT 6 (Covering the period

Section 6. PROCEDURES

11q

Volume I CATNUM

11r

APPENDIX II NIC DOCUMENT LOCATOR

12

INTRODUCTION 12a

The Locator consists of tables of contents for selected NIC documents. 12a1

The tables of contents extend to two or three levels. 12a1a

Through Locator the user can retrieve useful parts of documents online with a few commands. 12a1b

The headings in the tables of contents contain links to the corresponding online documents. 12a1c

The links include viewspecs that curtail print commands so they display table-of-contents-like samples of the object documents. 12a1d

In the syntax for commands below, CA is control-D and the text enclosed in square brackets is echoed on full-duplex terminals. 12a2

USING THE LOCATOR ONLINE 12b

To load locator: 12b1

Syntax: SP (nic,locator, CA *[!]* 12b1a

To print the introduction and instructions: 12b2

Syntax: p*[rint]* b*[ranch]* .1 CA *[!]* w CA *[!]* 12b2a

To list documents that you can reach with Locator, print branch .2 with viewspecs that show one line each of the first two levels and that show statement numbers. 12b3

Syntax: p*[rint]* b*[ranch]* .2 CA xbm CA 12b3a

To see the table of contents for a specific document, print the branch that names it with viewspecs set to show the level addressed + one more. 12b4

Syntax: p[rint] b[ranch] .STATEMENTNUMBER CA xeb CA
12b4a

To load the corresponding title or subtitle in the document itself, print the branch with an up arrow. 12b5

The up arrow searches the statement in locator for a link and then follows it. 12b5a

The system then prints the branch in the file named in the link. 12b5a1

Syntax: p[rint] b[ranch] .STATEMENTNUMBER SP ↑ CA [↓] CA [↓]
.
.
.
.
[directory,filename]CA 12b5b

When you use an address that loads a second file, the system echoes the directory and file name.
12b5c

If you want to find an item in a catalog or directory (locator branches .2d and .2e) the best method is to search by content. The result of the print branch command will inform you of the best way to search in each catalog. Control-o or rubout will stop the printing.
12b6

If you are searching in a document of normal text, select the heading of interest to you in the online document, and print the statement with the viewspecs set to display the complete text. 12b7

Syntax: p[rint] s[tatement] .STATEMENTNUMBER CA w CA
12b7a

To return from a file to which you linked to the current file, use the jump to file return command. 12b8

Syntax: SP & CA. 12b8a

The linkstack command will print out a list of the last five files you have loaded. 12b9

INDEX pages=11
(journal,7481,:x) 12c1n

NIC JOURNAL USER GUIDE pages=78 12c2

CONTENTS pages=5
(journal,7635,:x) 12c2a

INTRODUCTION pages=10
(journal,7636,:xb) 12c2b

THE JOURNAL pages=14
(journal,7637,:x) 12c2c

IDENTIFICATION SYSTEM pages=10
(journal,7638, :xb) 12c2d

NUMBER SYSTEM pages=5
(journal,7639,:xb) 12c2e

SUMMARY OF JOURNAL SYSTEM COMMANDS pages=5
(journal,7640,:xb) 12c2f

COMMAND SUMMARY pages=4
(journal,7640,1:xb) 12c2g

INDIVIDUAL IDENTS pages=12
(journal,7641,1:xb) 12c2h

GROUP IDENTS pages=1
(journal,7642,1:xb) 12c2i

AFFILIATION IDENTS pages=3
(journal,7643,:xr) 12c2j

INDEX pages=6
(journal,7644,:x) 12c2k

NIC TNLS EXERCISE FILES pages = 23 12c3

XED...tutorial file in line editing and structural editing pages=13
(nic,xed,1:wn) 12c3a

XPARCOP...tutorial file in manipulation of partial copies

EXTENDED DIRECTORY OF GROUPS [and members] pages=30
(nic,xtndgrp,entry:whDn) 12c5g

EXTENDED DIRECTORY OF INDIVIDUALS [with addresses] pages
=29
(nic,xtndid, entry:Dgn) 12c5h

ARPA NETWORK RESOURCES NOTEBOOK Pages=62 12c6

INDEX pages=23
(nic,resindex,xentry) 12c6a

BBN-TENEX pages=10
(nic,bbn-tenex,:x) 12c6b

CASE pages=5
(nic,case,:x) 12c6c

CARNEGIE pages=5
(nic,carnegie,:x) 12c6d

HARVARD-1 pages=3
(nic,harvard-1,:x) 12c6e

HARVARD-10 pages=7
(nic,harvard-10,:x) 12c6f

ILLINOIS pages=5
(nic,illinois,:x) 12c6g

INTRO pages=8
(nic,intro,:x) 12c6h

LL-67 pages=3
(nic,ll-67,1:x) 12c6i

LL-TX-2 pages=15
(nic,ll-tx-2,:x) 12c6j

MIT-AI pages=3
(nic,mit-ai,1:x) 12c6k

MIT-DMCG pages=7
(nic,mit-dmcg,:x) 12c6l

SRI-ARC 8 JUNE 1972 13041
Appendices
Typical Resource Notebook Entry

APPENDIX III A TYPICAL NETWORK RESOURCE NOTEBOOK ENTRY 13

UCSB Computation Center IMP #3
IBM 360/75 HOST #0 13a

I. Personnel 13b

Area Code is: 805 13b1

A. Administrator: 13b1a

Charles Loepkey 961-2261 13b1a1

B. Software: 13b1b

Ron Stoughton 961-3454 13b1b1

C. Hardware: 13b1c

Bob Ploger 961-2462 13b1c1

D. NIC Station Agent: 13b1d

Connie Rosewall 961-3221 13b1d1

E. NIC Technical Liaison: 13b1e

Ron Stoughton 961-3454 13b1e1

F. Principal Investigator: 13b1f

David Harris 961-2534 13b1f1

G. Operations Supervisor: 13b1g

Steve Neumann 961-2274 13b1g1

Mailing address is: 13b2

Computer Center
University of California at Santa Barbara
Santa Barbara, California 93106 13b2a

II. Installation Type 13c

The installation includes both research and service features. From approximately 9:00 am to 10:00 pm on weekdays and from 12:00 noon to 6:00 pm on Saturdays the Center provides batch service local and Online System service to both local and remote users. At other times, research and limited batch service proceed concurrently. Batch programs run under the OS MVT (multiprogramming with a variable number of tasks) operating system. Unit record equipment is under control of HASP (Houston Automatic Spooling Priority System). Online users run under an expanded version of the Culler-Fried system developed at UCSB. 13c1

III. Equipment 13d

A. The computer at this site is an IBM 360/75 with a memory size of 2,524,298 8-bit bytes, of which 2M bytes are 2361 core storage, and the remainder is 2365 processor storage. The /75 has a word length of 32 bits, but its instruction set is byte oriented. 13d1

B. Peripheral equipment includes: 13d2

1. 1 2540 card read/punch unit (1000 cpm read, 300 cpm punch) 13d2a

2. 2 1403 line printers (132 columns, 1000 lpm) 13d2b

3. 16 2314 disc drives (28M bytes each) 13d2c

4. 2 2415 magnetic tape drives (one 7-track, one 9-track) 13d2d

5. 1 digital incremental plotter 13d2e

6. 75 storage tube remote graphics terminals 13d2f

IV. Consoles 13e

An IBM 2701 Data Adapter Unit has been installed on the multiplexor channel which permits the 360/75 to communicate with a wide variety of remotely located terminals, devices, and processors. The terminals, devices, and processors served by the 2701 offer a wide range of transmission

methods, transmission speeds, transmission codes, line capacities, and application flexibility. At this writing no terminals are attached to the 2701. However, we plan to install either a TTY-37 or IBM 2741 in the near future which will serve as the Network Agent's reference and communication station. Other terminals will be added as user demand requires. 13e1

V. Physical Resources 13f

A. The Online System (OLS) supports a maximum of users concurrently. Network and local users will compete for use of OLS, with the added restriction that some maximum number of users from the Network will be allowed access to the System concurrently (this maximum number is currently ten, but will be increased if demand warrants). OLS is available for Network use according to the following schedule: (Note: see diagram from NIC #6808 "UCSB SYSTEM 360/75") 13f1

Mon	5:00 am to 10:00 pm
Tues	9:00 am to 10:00 pm
Wed	5:00 am to 10:00 pm
Thurs	5:00 am to 10:00 pm
Fri	9:00 am to 10:00 pm
Sat	12:00 noon to 6:00 pm

13f1a

In addition, Network users of OLS may run at other times when OLS happens to be up but the stability of the system is not guaranteed. Prime time for batch users is as follows: 13f2

Mon	8:00 am to 10:00 pm
Tues	9:00 am to 10:00 pm
Wed	8:00 am to 10:00 pm
Thurs	8:00 am to 10:00 pm
Fri	9:00 am to 10:00 pm
Sat	12:00 noon to 6:00 pm

13f2a

In addition, Network users of batch may run at other times on an irregular basis. The Computer Center is always open. 13f3

B. Initial experimental use of OLS can be conducted under a special user number. The relevant accounting parameters are specified in RFC#74 (NIC #5417). Other than experimental usage must be arranged with the Computer Center administrator and will be charged for at the then-current rates. Initial experimental use of batch services can be

conducted under a special account. The relevant accounting parameters are specified in RFC #105 (NIC #5775). Production runs must be arranged with the Computer Center administrator and will be charged for at the then-current rates. Computer Center rates are changed periodically in accordance with past usage and projected usage. Users holding valid Computer Center account numbers will be notified in advance of any change in the rate structure. The billing rates currently in effect are as follows: 13f4

Batch 13f4a

central Processor
$2.42356/minute 13f4a1

Core Storage
0.00023881/KCS 13f4a2

printer
0.753949/1000 lines 13f4a3

Card Reader
0.921726/1000 cards 13f4a4

Card Punch
3.07242/1000 cards 13f4a5

Channel Interrupts
3.16/1000 13f4a6

High Priority Service
25% surcharge 13f4a7

Online System 13f4b

Console Connect Time
$4.00/hour 13f4b1

Core Storage
0.1776/KBH 13f4b2

Computer Resource Units
0.00011863/CRU 13f4b3

Disk Storage
0.10/KBM 13f4b4

Printer Form	13f4c
Two part $0.012/page	13f4c1
Three part 0.019/page	13f4c2
Four part 0.025/page	13f4c3
Five part 0.033/page	13f4c4
Six part 0.040/page	13f4c5
Cards	13f4d
(2000/box) $2.25/box	13f4d1
Labels	13f4e
(three across, 36 labels/page) $0.06/page	13f4e1
Disk Packs	13f4f
storage $2.00/month	13f4f1
2316 - rental 14.50/month	13f4f2
Magnetic Tapes	13f4g
2400 ft. - purchase $18.00/each	13f4g1
2400 ft. - rental 1.50/month	13f4g2
1200 ft. - purchase 13.00/each	13f4g3

storage - any size
0.50/month 13f4g4

C. The NCP provides every user with a 256-byte buffer for temporarily queuing incoming or outgoing data. When the resources of a local receiving process are sufficient, the NCP will - using the Host-Host protocol mechanism - allow the connected, foreign process to transmit maximum length messages (8095 bits). When a local sending process presents to the NCP with a single system call a sufficiently large amount of data to be output, the data will be transmitted as one or more maximum length messages. 13f5

D. The Computer Center will support third level direct access storage by providing a simple file storage and retrieval process. The amount of online storage provided will depend upon the demand and availability of disc drives. An initial allocation of 29M bytes is planned. Files so stored will be backed up to magnetic tape daily. The back-up tape(s) will be offline and available only in case the online copies are destroyed. An exact rate schedule has not been established for this facility, but a billing rate similar to that used for OLS long term storage (see paragraph B) can be expected. 13f6

VI. Interests and Capabilities 13g

The UCSB Computer Center provides batch service to on- and off-campus users, and Online System service at approximately 55 on-campus and 20 off-campus terminals. Much of UCSB's research effort has been directed toward development of its Online System. 13g1

VII. Login 13h

Specifications for logging into OLS through the Network are contained in RFC #74 (NIC #5417). The most recent user's manual for OLS is on file with the NIC. Specification for Network submission of batch jobs and for retrieval of the resulting 'printed' output are contained in RFC #105 (NIC #5775). The software listed in Section X can be invoked for batch processing using the appropriate job control language (JCL). 13h1

VIII. Computer Operator 13i

Communication with the operator through the Network is not possible. The operator may always be reached by phone at (805) 961-2274. 13i1

IX. Miscellaneous 13j

None 13j1

X. Programs 13k

The only processes presently accessible to Network users are OLS and RJE which have been documented in RFC #74 (NIC #5417) and RFC #105 (NIC #5775) respectively. A user manual describing OLS in detail is on file at NIC and available to Network users. A list of available batch-mode software begins below. 13k1

A. FORTRAN IV (IBM: levels G & H) (University of Waterloo: WATFOR, WATFIV) - a high-level language oriented toward mathematical problems. 13k1a

B. PL/1 (IBM: level F) (Cornell University: PLC) - a high-level, general purpose language. 13k1b

C. PLOT (UCSB) - a package callable from FORTRAN and PL/1 programs which allows display of graphical data on a digital plotter. 13k1c

D. RPG (IBM) - a language for generation of business-type reports. 13k1d

E. ASSEMBLER (IBM: level F) (University of Waterloo: level G) - a low-level language for systems programmers. 13k1e

F. SSP (IBM) - a scientific subroutine package for FORTRAN and PL/1. 13k1f

G. GPSS (IBM) - a high-level language oriented towards the social sciences. 13k1g

H. SPSS (Stanford) - a set of statistical routines oriented towards the social sciences. 13k1h

I. SORT/MERGE (IBM) - a program for sorting and merging data sets. 13k1i

J. BIOMED (UCLA) - a set of statistical routines for FORTRAN users. 13k1j

K. UTILITIES (IBM) - a set of programs for the manipulation of data sets. 13k1k

L. OSIRIS-II (University of Michigan) - an organized set of integrated routines for investigation with statistics. 13k1l

M. COBOL (IBM: level F) - a high-level language oriented toward business problems. 13k1m

N. ALGOL (IBM: level F) (Stanford University: ALGOL-W) - a high-level language oriented toward mathematical programs. 13k2

O. CSMP (IBM) - a high-level language oriented toward modeling problems. 13k3

P. SNOBOL (Bell Labs) - a string manipulation language. 13k4

Q. XTAB/FREQ (UCSB) - programs for cross tabulation and frequency count. 13k5

APPENDIX IV NIC FUTURE SOFTWARE PLANS 14

INTRODUCTION 14a

What follows is a compilation of the ARC software tasks that are foreseen as important to the NIC, and hence for which the NIC should assume partial or total responsibility. 14a1

NETWORK 14b

MOTIVATION and SOFTWARE REQUIREMENTS: 14b1

To exploit the Network -- via forthcoming, Network-standard protocols -- in delivering the NIC's sevices to the Network community. 14b1a

SPECIFIC PROJECTS: 14b2

Inter-host File Transfers: 14b2a

To hosts' file systems 14b2a1

MOTIVATION: 14b2a1a

To enable remotely-generated text files to be entered into the NIC for: 14b2a1a1

Manipulation via NLS 14b2a1a1a

Retrieval of files previously archived into the Net 14b2a1a1b

Remote submission of Journal entries 14b2a1a1c

To enable the transmission of NIC files to remote hosts for: 14b2a1a2

Return of NLS-manipulated files 14b2a1a2a

Manipulation by remote text editor 14b2a1a2b

Storage at remote host 14b2a1a2c

Archiving of ARC files 14b2a1a2d

Trouble-shooting is inevitably required at the remote host, for each of the above activities 14b2c

NLS 14c

System Development 14c1

MOTIVATION: 14c1a

NIC has an obvious interest in promoting the continued development of NLS, the primary tool which it offers its users. 14c1a1

Although the NIC is presently primarily concerned with those system changes which benefit TNLS (since TELNET service is all that's currently provided on a supported basis), the expected offering of DNLS to the network community motivates the NIC to interest itself in that version of the system as well. 14c1a2

SOFTWARE REQUIREMENTS: 14c1b

Take an active part in NLS development, sharing in the software load, and assume implementation responsibility for those features which primarily benefit users of the NIC. 14c1b1

SPECIFIC PROJECTS: 14c1c

New, TNLS-specific features 14c1c1

Novice thru expert modes 14c1c1a

MOTIVATION: 14c1c1a1

To isolate the novice user from advanced concepts which would only confuse him, while providing the sophisticated user with acess to the full capabilities of the System. 14c1c1a1a

SOFTWARE REQUIREMENTS: 14c1c1a2

Stratify TNLS in such a way that levels of capability consistent with the user's expertise can be provided, 14c1c1a2a

The impending, wholesale revision of NLS syntax may be just the vehicle for realizing this need. 14c1c1a2b

Lingering statement numbers 14c1c1b

MOTIVATION: 14c1c1b1

To reduce the frequency with which the user must regenerate his display (a time-consuming thing to do from a TTY) to keep track of statement-number changes 14c1c1b1a

SOFTWARE REQUIREMENTS: 14c1c1b2

Integrate into TNLS, some of the capabilities inherent in DEX 14c1c1b2a

Line-drawing construction 14c1c1c

MOTIVATION and SOFTWARE REQUIREMENTS: 14c1c1c1

To provide a mechanism by which line drawings can be constructed from a TTY-like terminal for later display in DNLS 14c1c1c1a

New, DNLS-specific features 14c1c2

Shared display screens 14c1c2a

MOTIVATION: 14c1c2a1

To promote the real-time cooperation of NIC users in the construction and examination of NLS files 14c1c2a1a

SOFTWARE REQUIREMENTS: 14c1c2a2

Provide a mode of operation in which a user's display can be replicated on another terminal, and the bug positions of each user displayed on both screens 14c1c2a2a

Graphics 14c1c2b

MOTIVATION: 14c1c2b1

permit a generalization of NLS' current tree structure. 14c1c3f2a1

The Modular Programming System (MPS), which will ease the task of checking out new system components and of monitoring system behavior. 14c1c3f2a2

Miscellaneous 14c1c3g

MOTIVATION and SOFTWARE REQUIREMENTS: 14c1c3g1

Implement such features as addressing by content, the saving of viewchange-viewspec information, and whatever additional features prove to be necessary or desirable. 14c1c3g1a

System Maintenance 14c2

MOTIVATION and SOFTWARE REQUIREMENTS: 14c2a

The NIC has an obvious responsibility for and interest in participating in the maintenance of NLS. 14c2a1

This responsibility includes activities which: 14c2a2

Locate and fix bugs 14c2a2a

Reduce the cost to the user of using the System 14c2a2b

Decrease response time by inproving code efficiency 14c2a2c

Improve reliability 14c2a2d

SPECIFIC PROJECTS: 14c2b

statistics gathering 14c2b1

MOTIVATION and SOFTWARE REQUIREMENTS: 14c2b1a

To obtain information concerning: 14c2b1a1

The cost of each NLS command to help locate those points in the software which should be made more efficient. 14c2b1a1a

The frequency with which specific commands are used, and then to evaluate the worth of supporting and maintaining infrequently used commands. 14c2b1a1b

The difference in system overhead between Network and local users to evaluate the performance of Network-related code in the Monitor 14c2b1a1c

The effect of changes in the amount of core available to the system upon system performance, and if appropriate to recommend changes in configuration of the system. 14c2b1a1d

Resource allocation control 14c2b2

MOTIVATION and SOFTWARE REQUIREMENTS: 14c2b2a

To implement mechanisms for controlling the allocation of such system resources as CPU time and secondary storage among local and Network users. 14c2b2a1

Literal collection and feedback by the Monitor 14c2b3

MOTIVATION and SOFTWARE REQUIREMENTS: 14c2b3a

To evaluate the effect upon system performance of moving responsibility for literal collection and feedback from NLS to the Monitor, and if the effect is found to be significant, to implement the change. 14c2b3a1

Augmentation of secondary storage 14c2b4

MOTIVATION: 14c2b4a

To increase the amount of secondary storage available to house the NIC data-base. 14c2b4a1

SOFTWARE REQUIREMENTS: 14c2b4b

In addition to the possibilities already described for archiving files in the Net, to

APPENDIX V A LIST OF TNLS COMMANDS

15

A list of currently available TNLS commands follows. They are described in detail in the TNLS User Guide. (See 7470). 15a

Append statement
Break statement
Copy entity
Delete entity
Execute
    Assimilate
    Browse
    Catalog Numbers
    Device Specification
    Edit
    File Verify
    Identification System
    Insert Sequential
    Journal
    Logout
    Marker
    Name Delimiters
    Ownership
    Quit
    Reset
    Status
    Unlock File
    Viewchange
Fix Marker
Goto
    Baseline
    Exec
    Merge
    Programs
    Sort
    Use measurement
Insert entity
Load File
Move entity
Null File

Output
    File
    Quickprint
    Device
        Dura
        FR80
        Printer
        Sequential
        Teletype
    Compiler
    Assembler
    Sequential
Print
    statement
    branch
    plex
    group
Replace entity
Substitute
Transpose entity
Update File
Viewspecs
Xset
. -- Show point
; -- Comment Command
Give context commands
    \ -- Backslash Command
    / -- Slash Command
↑ -- Up Arrow Command, jump to back

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)*
Augmentation Research Center
Stanford Research Institute
Menlo Park, California 94025

2a. REPORT SECURITY CLASSIFICATION
UNCLASSIFIED

2b. GROUP

3. REPORT TITLE
ONLINE TEAM ENVIRONMENT

4. DESCRIPTIVE NOTES *(Type of report and inclusive dates)*
Final Technical Report 9 February 1971 -- 9 May 1972

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

6. REPORT DATE
8 June 1972

7a. TOTAL NO. OF PAGES
266

7b. NO. OF REFS
76

8a. CONTRACT OR GRANT NO.
F30602-70-C-0219

b. PROJECT NO.
ARPA Order No. 967

c.

d.

9a. ORIGINATOR'S REPORT NUMBER(S)
13041

9b. OTHER REPORT NO(S) *(Any other numbers that may be assigned this report)*
RADC-TR-72-232

10. DISTRIBUTION STATEMENT
Approved for public release; distribution unlimited

11. SUPPLEMENTARY NOTES
Monitored by
RADC/ISIM, Duane Stone
Griffiss AFB, New York 13440
AC 315 330-3857

12. SPONSORING MILITARY ACTIVITY
Defense Advanced Research Projects Agency
Arlington, Virginia 22209

13. ABSTRACT

During the period covered, our computer system became operational on a PDP-10 computer with a TENEX Timesharing System. The Information Center ARC maintains for the ARPA computer Network became much more active both in distributing Network documentation and in supporting distributed dialog among experimenters on the Network. To our online system we added a number of features that extend the power of the user including a way of writing and calling special purpose programs, cross-file editing, and spooling input on magnetic tape for later entry into the system.

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
Augmented human intellect						
Information centers						
Information retrieval						
Indexes						
Document Storage						
Man-machine systems						
Data displays						
Text processing						
Networks						
ARPA network						