ONLINE TEAM ENVIRONMENT. (NETWORK INFORMATION CENTER AND COMPUTER AUGMENTED TEAM INTERACTION)

Douglas C. Engelbart

Stanford Research Institute

Prepared for:

Rome Air Development Center
Defense Advanced Research Projects Agency

8 June 1972
Best Available Copy
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.
<table>
<thead>
<tr>
<th>KEY WORDS</th>
<th>LINK A</th>
<th>LINK B</th>
<th>LINK C</th>
</tr>
</thead>
<tbody>
<tr>
<td>Augmented human intellect</td>
<td>ROLE</td>
<td>WT</td>
<td>ROLE</td>
</tr>
<tr>
<td>Information centers</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Information retrieval</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Indexes</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Document Storage</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Man-machine systems</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Data displays</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Text processing</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Networks</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ARPA network</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
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

Approved for public release: distribution unlimited.

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

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

Approved for public release; distribution unlimited.

This research was supported by the Defense Advanced Research Projects Agency of the Department of Defense and was monitored by E. L. Stone, RADC (IS1W), OAFB, NY 13440 under Contract F30602-70-C-0219.
PUBLICATION REVIEW

This technical report has been reviewed and is approved.

[Signature]
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 P. Ausubel, Walter L. Beas,
William J. Davall, Douglas C. Engelbart,
William R. Ferguson, Ben A. Hardeman,
Marty E. Hardy, J. David Hopper,
Charles M. Irby, Mildred R. Jerwigen,
Diana S. Kaye, Michael D. Kadlick,
Linda L. Lane, Harvey G. Lehmkuhle,
Donald Limati, Priscilla M. Lister,
N. Dean Meyer, Jeannie B. North,
James C. Norton, Cynthia Page,
Bruce L. Parsley, William F. Paxton,
Jeffrey C. Petters, Ralph Prather,
Jake Retliff, Paul Rech, Barbara E. Row,
Jacques F. Vallée, Edwin K. Van De Riet,
Dirk H. van Neswy, Kenneth E. Victor,
Donald C. Wallace, Richard W. Watson,
and James E. White

In addition, the following consultants:

Don L. Andrews,
James A. Fadiman

and the following former members of the staff:

Geoffrey D. Bell, Roger D. Bates,
Vernon R. Daughman, Mary G. Caldwell,
Roberta A. Carillon, David O. Casseres,
Donald Cone, Mary S. Church, Robert L. Dendy,
William E. English, Ann R. Geoffrion,
Jared H. Harris, John T. Melvin,
Martha E. Trundy, and John W. Yarborough
Abstract

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

TENEX

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.

Network Information Center

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.

Hardware

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

New features in NLS

To our online system we have added:

- A command language, DEX, which allows entry of text on tape for later automatic processing into NLS files;
- 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;
- Cross file editing to our typewriter-oriented command language, NLS, along with other features that suit NLS to typewriter terminal use; and
- To our display system, the capacity to split the screen, load several files at once, and transfer information from one file to another.

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.

MANAGEMENT SYSTEMS

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 A4C.
CONTENTS

<table>
<thead>
<tr>
<th>Content</th>
<th>Statement Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABSTRACT</td>
<td>1</td>
</tr>
<tr>
<td>CONTENTS</td>
<td>2</td>
</tr>
<tr>
<td>SUMMARY</td>
<td>3</td>
</tr>
<tr>
<td>Introduction</td>
<td>3a</td>
</tr>
<tr>
<td>Highlights of 1970</td>
<td>3b</td>
</tr>
<tr>
<td>Highlights of 1971</td>
<td>3c</td>
</tr>
<tr>
<td>References</td>
<td>3d</td>
</tr>
<tr>
<td>TEAM AUGMENTATION</td>
<td>4</td>
</tr>
<tr>
<td>Journal</td>
<td>4a</td>
</tr>
<tr>
<td>Handbook</td>
<td>4b</td>
</tr>
<tr>
<td>Baseline Records System</td>
<td>4c</td>
</tr>
<tr>
<td>Basic NIS</td>
<td>4d</td>
</tr>
<tr>
<td>Internal Organization</td>
<td>4e</td>
</tr>
<tr>
<td>References</td>
<td>4f</td>
</tr>
<tr>
<td>NETWORK INFORMATION CENTER: DEVELOPMENT AND OPERATIONS</td>
<td>5</td>
</tr>
<tr>
<td>Introduction</td>
<td>5a</td>
</tr>
<tr>
<td>The NIC Public</td>
<td>5b</td>
</tr>
<tr>
<td>Present NIC Services</td>
<td>5c</td>
</tr>
<tr>
<td>Relation of the NIC to the Augmentation Research Center (APR)</td>
<td>5d</td>
</tr>
<tr>
<td>Operations</td>
<td>5e</td>
</tr>
<tr>
<td>Online Services</td>
<td>5f</td>
</tr>
<tr>
<td>Offline Services</td>
<td>5g</td>
</tr>
<tr>
<td>Experience Using the ARPANET</td>
<td>5h</td>
</tr>
<tr>
<td>Conclusion</td>
<td>5i</td>
</tr>
<tr>
<td>References</td>
<td>5j</td>
</tr>
<tr>
<td>NETWORK PARTICIPATION</td>
<td>6</td>
</tr>
<tr>
<td>Introduction</td>
<td>6a</td>
</tr>
<tr>
<td>Protocol Development</td>
<td>6b</td>
</tr>
<tr>
<td>Network Coordination</td>
<td>6c</td>
</tr>
<tr>
<td>References</td>
<td>6d</td>
</tr>
</tbody>
</table>

Online Team Environment

3
**Contents**

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>COMPUTER FACILITY</strong></td>
<td>7</td>
</tr>
<tr>
<td>Hardware</td>
<td>7a</td>
</tr>
<tr>
<td>Introduction</td>
<td>7a1</td>
</tr>
<tr>
<td>Present Configuration</td>
<td>7a2</td>
</tr>
<tr>
<td>Problems We Have Been Facing</td>
<td>7a3</td>
</tr>
<tr>
<td>System Software</td>
<td>7b</td>
</tr>
<tr>
<td>Immac Support for DNLS</td>
<td>7b1</td>
</tr>
<tr>
<td>TEXAC</td>
<td>7b2</td>
</tr>
<tr>
<td>Superwatch</td>
<td>7b3</td>
</tr>
<tr>
<td>References</td>
<td>7c</td>
</tr>
<tr>
<td><strong>PLANS</strong></td>
<td>8</td>
</tr>
<tr>
<td>Goals</td>
<td>8a</td>
</tr>
<tr>
<td>Service to Users</td>
<td>8b</td>
</tr>
<tr>
<td>Basic Project Work</td>
<td>8c</td>
</tr>
<tr>
<td>References</td>
<td>8d</td>
</tr>
<tr>
<td><strong>GLOSSARY</strong></td>
<td>9</td>
</tr>
<tr>
<td><strong>APPENDIXES</strong></td>
<td>10</td>
</tr>
<tr>
<td>Handbook Contents</td>
<td>11</td>
</tr>
<tr>
<td>Document Locator</td>
<td>12</td>
</tr>
<tr>
<td>A Typical Network Resource</td>
<td>13</td>
</tr>
<tr>
<td>Notebook Entry</td>
<td>14</td>
</tr>
<tr>
<td>VIC Software Plans</td>
<td>14</td>
</tr>
</tbody>
</table>

Online Team Environment

4
SUMMARY

INTRODUCTION

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.

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, KADC-TR-71-175, AN 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.

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

"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 number printed right. Online, a reader may cite such an address and move automatically to the appropriate part of the report.

A four- or five-digit number in parenthesis (e.g., (0277)) 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.

A glossary appears in (.9).

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

The detailed accounts are in the following Handbook documents:

- TREE META (10857),
- MEX USERS GUIDE (9934),
- NLS PRELIMINARY USERS GUIDE (10703),
- MTC NLS USERS GUIDE (7470),
- MTC JOURNAL USERS GUIDE (7635),
- TLO PROGRAMMING GUIDE (USER GUIDE) (9246),
- TLO - A Programming Language for the Augmentation Research Center (Systems Programmer's Guide) (7052)

HIGHLIGHTS OF 1970

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.

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.

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.
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.

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.

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.

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).

In using the Net to re-program our PDP-10 we typically sent blocks to UXAD 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.

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...
SRI-ARC 8 June 1972 13041

Summary
Highlights of 1970

Institution. We organized with the other sites of 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-set document numbering system.

In the spring of 1970 we decided that DEC's PDP-10 with associated software and paging box from MHN 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 MHN 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.

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, SRI, 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 could load and edit independently. Many other features such as Mail, Journal, calculator were substantially improved in the transfer.

 Highlights of 1971

Team Augmentation

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.

Online Team Environment
Dialog Support System

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.

ARC and Network personnel use the Journal system daily.

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

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

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 the list of recipients the user spells out.

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

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

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

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

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.

These statements contain a link to the document, along with the sender's identification,
A message facility has been incorporated in the "DP-10 Journal," which eliminates the mail system used on the XDS-940.

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

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

Handbook

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.

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

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.

Baseline Record System

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.

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.
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.

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.

There are now 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.

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.

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.

We are not satisfied with the present Baseline Record system.

We feel that our ARC users were not well guided and trained in RPS use and

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

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.

**Basic NLS**

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.

**TNLS and DEX**

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 version of an offline mode (DEX) has been introduced.

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

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

in DNLS, split screens allow the user to view any two passages and control cross-file editing visually.
Viewspect make possible selective assimilation of information from one file into another.

New special purpose subsystems have been developed or improved.

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

Language development has continued.

At present the primary language systems developed and in use at APC are the Tree-Meta Compiler-compiler System and the L1O Programming language system which was written in Tree-Meta.

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

Internal Organization

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.

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.

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.

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

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

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

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

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

PODAC institutionalizes continuing personal and organizational development.

Network Information Center: Operations and Development

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.

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.

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

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

The NIC is not a classical information center because
it provides a wider range than bibliographic and library services.

The NIC Public

One of the problems in the design of an information service is to determine the clientele and its needs. Our initial analysis showed us four main needs:

1. Reference and General Network Information;
2. Collaboration Support;
3. Document Handling and Creation; and
4. Training.

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.

NIC Services

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

Online:

1. Access to the typewriter version (TNLS) of the Augmentation Research Center Online System (NLS) for communique creation, access, and other, experimental use.
2. Access to Journal, Number, and Identification Systems which allow messages and documents to be transmitted to Network participants.
3. Access to a number of online information bases through a special Locator file using NLS link mechanisms.

Online Team Environment
Offline:

(i) A Network Information Center Station set up at each site with:
   (a) A Station Agent to aid in use of the NIC.  
   (b) A Liaison to provide technical information about his site.  
   (c) A Station Collection containing a subcollection of documents of interest to network participants.

(2) Techniques for gathering, producing and maintaining NIC functional Documents, such as:
   (a) Current Catalog of the NIC Collection.
   (b) ARPA Network Resource Notebook.
   (c) Directory of Network Participants.
   (d) NIC User Guide.

(3) General network referral and handling of document requests.

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

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

(5) Crude selective distribution to Station Collections.

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

NIC Goals

In the course of its evolution, the ARPANET will
continue to generate needs for new software services in interactive data management.

we propose to develop a user-oriented information facility based upon the NLS system and initially serving the needs identified in (3c2a). 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.

Network Participation

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.

Computer Facility

Hardware

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 (6277,).

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.

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 IBM network interface and a new DEC 8P-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.

System Software

TENEX

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.

within the system:

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

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

We have made many changes to the teletype routine to accommodate our displays.

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

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.

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.
In the User's View

We have set up an advise command so one terminal may control a job loaded at another terminal.

We have added routines that log out a user who does nothing for a certain time, and that refuse entry if the system is overloaded.

SUPERWATCH

To help find out what is going on within our timesharing system we have developed an information gathering and formatting program called Superwatch.

In general Superwatch has been valuable:

To verify that the system is working as designed.

To identify the cause of poor service at the time it is happening (e.g. a bug, hardware malfunction, or just overloading).

To identify the "weak link" in the system configuration (drum, disk, memory or CPU capacity).

To evaluate changes in the system or hardware configuration.

Plans for the Future

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:

Advancing the techniques available to ARC and Network system builders and users for augmenting the development and application of computer-ted information systems.

Making the Network Information Center into both:
Summary

Highlights of 1971

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

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

And moving useful augmentation techniques and services out into the ARPA Network Community.

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

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

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

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

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

and wherein we become ever better at

(4) Operating a utility service.

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.

REFERENCES

Summary References


TEAM AUGMENTATION


JOURNAL

Introduction

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.

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.

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:

Perhaps he was there at Time T, but 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.

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.

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.

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.

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.

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.

Here, the communication between Persons A and D 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.

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 these records. The ARC Journal is the central feature of this intragroup documentation system.

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.
Our DSS will provide the following general online aids: multiwindowed displays; simultaneous and independent mobility and view control among many files; link-set-up 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.

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.

The PDP-10 Journal

During 1971, implementation of the initial PDP-10 Journal system was completed.
The PDP-10 Journal system provides for automated entry of Online documents in contrast to the essentially manual technique used on the XDS-910.

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.

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.

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.

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

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

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

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

A message facility has been incorporated in the PDP-10 Journal, which replaces the mail system used on the ADS-910.

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

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

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.

**User appearance**

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, Flex, or file) or as a message (literal) to be typed in.

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

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.

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

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

**Clerk:** Person actually submitting the document.
comments: A comment which is kept in the document header as an appendage to the document.

distribution: A list of persons or groups to receive copies of the document.

keywords: Key words which may be used for document retrieval at a later time.

obsoletes: A list of documents obsoleted by the document being submitted.

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

The subcollections listed here are in addition to:

Any subcollections associated with the submitter by default.

Any groups included in the distribution list.

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

updates: A list of documents updated by the document.

Additional to the parameter specification commands are:

commands for control

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

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

status Command: Shows the current status of the entry parameters.

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.
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.

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.

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

The user is then returned to the NLS command mode.

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

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.

Identification System

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

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.

The outgrowth of this need is the Identification system.

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.

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

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

The Identification system is used extensively by all
chases of the Journal.

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

Number system

The Number system provides a capability for centrally
assigning Master catalog and Network Working
Group/Request for Comments (KGC/KFC) numbers.

There is a set of KLS commands for directly assigning
catalog numbers, and for pre-assigning KFC and Journal
numbers.

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

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

Document Access

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

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

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

while the master collection is maintained in its
original form, documents from the access collection
may be checked out, annotated, and copied by KFC
personnel.

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

unline Team Environment
As indispensable aids to the user, ARC provides author, number, and titleword indices.

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

**ARC JOURNAL INDEX BY AUTHOR**

<table>
<thead>
<tr>
<th>Title</th>
<th>Date</th>
<th>Number</th>
<th>Author</th>
</tr>
</thead>
<tbody>
<tr>
<td>The TEXEX Scheduler</td>
<td>21 Jul</td>
<td>7k19</td>
<td>Andrews</td>
</tr>
<tr>
<td>response memo</td>
<td>21 Jul</td>
<td>7k15</td>
<td>Andrews</td>
</tr>
<tr>
<td>response error</td>
<td>12 Jul</td>
<td>7394</td>
<td>Andrews</td>
</tr>
<tr>
<td>ref title</td>
<td>14 May</td>
<td>6532</td>
<td>Andrews</td>
</tr>
</tbody>
</table>
| known performance problems                                          | 28 Feb | 9311   | ARC Systeam Mas  
| SHEL problem                                                        | 10 Feb | 9331   | ARC Systeam Mas  
| Don Heimhoelter Proposal                                             | 13 Jul | 11041  | Auercbach |
| reply to 10951 (reply to 10953)                                     | 10 Jul | 11002  | Auercbach |
| reply to your comments on R10 documentation                           | 5 Jul  | 10533  | Auercbach |
| ref: update                                                          | 5 Jun  | 10633  | Auercbach |
| let me tell you about the Handbook                                   | 23 May | 10644  | Auercbach |
| ref: attacch-tion and at vs. after                                   | 11 May | 10657  | Auercbach |
| ref: output processor directives (again)                             | 11 May | 10613  | Auercbach |
| header / positioning options                                         | 8 May  | 10381  | Auercbach |
| ref: <documentation>request                                          | 1 May  | 10332  | Auercbach |

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.

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 currently 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

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.

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

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

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

Copies of any document in the Journal collection are distributed in a like manner using the Secondary Distribution command.
A user may specify the manner in which Journal documents addressed to himself are to be distributed.

Current delivery options are hardcopy and online.

A user may specify either or both of these options.

Other options will be provided as they become necessary.

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

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

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.

Online delivery is done by a background processor which is automatically started when TENEX is initiated.

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.).

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.

Special Features

Certain applications of the Journal system have required special handling.
Most notable of these special applications has been the
network working Group Request For Comments (NWG/RFC).

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.

This greatly facilitates the processing and distribution of
these documents.

Problems and Comments

Reliability

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

one of the major problem areas has, correspondingly, been
deleted manipulation, specifically file integrity.

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

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

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-safe in this area.

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 locking
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 NIS command level.

Operations

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

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 RPO2 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 graciously.

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

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 impossible.

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

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

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.

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

Future efforts will attempt to improve the efficiency.

Other systems and procedures within the AHC and Network environments are interfacing with the Journal system.

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

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

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

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

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.

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.
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.

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:

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

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.

At present, we have just finished the first and very primitive phase of organizing and obtaining in hard copy much documentation relevant to the contents of an ARC Handbook. An outline is included in this report. 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 hard copy 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.
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.

The Handbook as it now exists is by no means inclusive as its primary source is the Journal for information about system features, APC 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.

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

BASELINE RECORD SYSTEM

Introduction

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. 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.

Planning requires a framework within which information about goals, needs, possibilities, resources, and related dialog can be recorded, studied, and modified usefully. ARC planning and task activity is currently conducted in the LINAC operational framework outlined below, see (--,le3). 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.

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.

The basic objectives of the Baseline Record System are:

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.

Online Team Environment
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:

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).

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.

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.

Present Baseline Record System

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.

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.

Baseline Task Estimates as of 26 APR 72

<table>
<thead>
<tr>
<th>Month</th>
<th>Apr--May--Jun--Jul--</th>
</tr>
</thead>
<tbody>
<tr>
<td>QWR Annual Report 6622</td>
<td>XXX</td>
</tr>
<tr>
<td>Stackline Chairs</td>
<td>XXXX</td>
</tr>
<tr>
<td>Develop PAC ARC Baseline</td>
<td>XXXXXX</td>
</tr>
<tr>
<td>Projection TV</td>
<td>XXXXXX</td>
</tr>
<tr>
<td>Demonstration Training</td>
<td>XXXXXX</td>
</tr>
<tr>
<td>Resource Accounting Design</td>
<td>XXXXXX</td>
</tr>
<tr>
<td>Operations Development</td>
<td>XXXXXX</td>
</tr>
<tr>
<td>MIC Operations Coordinator</td>
<td>XXXXXX</td>
</tr>
<tr>
<td>User Documentation Maintain</td>
<td>XXXXXX</td>
</tr>
<tr>
<td>Management and Coordination</td>
<td>XXXXXX</td>
</tr>
<tr>
<td>Accounting</td>
<td>XXXXXX</td>
</tr>
<tr>
<td>Visitors</td>
<td>XXXXXX</td>
</tr>
<tr>
<td>Vacations</td>
<td>XXXXXX</td>
</tr>
<tr>
<td>Needs*Possibilities</td>
<td>XXXXXX</td>
</tr>
<tr>
<td>RIMS WP</td>
<td>XXXXXX</td>
</tr>
<tr>
<td>CAL Data Recent Revision</td>
<td>XXXXXX</td>
</tr>
<tr>
<td>RIMS Entry Conventions</td>
<td>XXXXXX</td>
</tr>
<tr>
<td>External Collaborator Coord</td>
<td>XXXXXX</td>
</tr>
</tbody>
</table>

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

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. 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.
Higher-Level Planning Needs

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.

Plan needs

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

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.

Planners needed and who makes them

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.

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 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.

Plan elements
The following eight items are basic considerations owners will provide in their plans:

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.
   ROUNH 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.

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

Comments on our Experience with the BRS to Date

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

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

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.

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.

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

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

Basic NLS User Features

Introduction

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.

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.

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.

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

- 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.

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.

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.

Such tools include:

- 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. 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.

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.

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.

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).

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.

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.

The system, as with all systems developed at APC, 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.

A new TNLS guide has been written (see -- 7h70), 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.

Several training courses for Network users of the NLS and TNLS have been held. They are described in this report as part of NLS activities (see--5610a).

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.

TNLS -- Technical overview

Introduction

The current implementation of NLS on the PDP-10 is a
This section presents an overview of the organization of NLS and the structure of files in the system.

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.

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

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 Flex 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 16. 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--.ld2f).

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

DNLS: See DNLS user guide. (10703.)

TNLS: See TNLS user guide. (7470.)
The format and structure of NLS files were determined by certain design considerations. 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 satisfies 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.
An NLS file is made up of a header and up to a fixed number (currently 165) of 512-word file blocks.

File Header

File header contents:

- 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

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

Ring element contents:

- Pointer to first successor statement.
- Pointer to successor statement.
- Pointer to the SDB that contains text for this statement.
- Display 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
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.

Statement Data Block (SDB) header contents:

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

String Identifiers and Text Pointers

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

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.

Locking mechanism -- Partial copies

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.

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.

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

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

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

Introduction

Interaction support

Terminal interaction support

Display interaction support

The display interaction support routines take input from display users, support various LIO display input constructions which allow the creation of simple interaction statements, and control the command feedback line, name area, view spec area, and box 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

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

Subsystem control

Command specification

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

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

Display control

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**

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

**Hardcopy formatters**

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**

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 view 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 --,ldle: for a discussion of the sequence generator with user programs.

User filters and reformatters

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

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

Editing

File manipulation algorithms

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

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

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

Inserting and outputting sequential files

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

Compilers

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, (9216,10) and see "ldleal" in this report.

IMOL, a procedure-oriented, block structured language which produces code for the IMILAC 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 "ld2e" of this report.)

MPL, the Modular Programming Language, an experimental new language to be used to rewrite NLS. (See "ld2f")

Text is passed to these compilers through the

Online Team Environment

67
sequence generator (and thus can be filtered and reformatted enroute to these various processors).

Utility routines

NLS file system

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 SUB's.

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

Statement name lookup.

NLS string system

Supports string manipulation constructs in the LI0 language and deals with the NLS Statement Data Block and Ring Block structure.

Miscellaneous support routines

Basic LI0 language support routines.

Call mechanisms.

Display support routines

Information writing on the screen.

Manipulating information on the screen.
Basic input routines

Basic typewriter terminal output routines

NLS -- New features

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

Name Delimiters

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.

Jump to word/Content

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

Null File

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.

Output Assembler

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

Output Compiler

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

Output: Sequential

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

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

Output Quickprint

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.

Update File -- File Locking

The NLS file system under TENEX provides a locking mechanism, which protects against inadvertent 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.

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.

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.

Online Team Environment
Goto Exec

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.

Execute Logout

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.

The following features in DNTS are new on the PDP-10:

- split Screen and Cross File Editing
- Display Screen Splitting and Formatting

FIGURE 19. (Opposite) Overall NLS logical structure.

Goto Display Area Control

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 GA 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
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 B'G) is moved to the new position (second B'G). 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 or 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 DMIS**

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

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
FIGURE 19  OVERALL NLS ORGANIZATION
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 raised, the display area containing the cursor receives the new viewspecs.

Substitute Commands: Change

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

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.

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.

Display Creation Efficiency Improvement

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.

The Deferred Execution System (DEX)

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

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.

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.

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.

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

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.

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.

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.

"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."
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.

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 mode.

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.

A manual for DEX-1 is currently available. (9934).
The design for DEX-2 is documented in (924),.

Other subsystems

Sorter-Merge-Updater Description

General Implementation Description

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 LIO programs compiled with the "Goto Programs LIO User Program Compile" command.

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 LIO 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:

Key procedure:

Sort key procedures are written in the LIO 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

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

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. 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.

**Output Processor Addition**

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 the use of the FR-80 microfilm device.

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

**FR-80 Output Processor Device**

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:

- 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**

**Introduction**

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.

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 (5277). 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.

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

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 contains 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.

The Network Information Center at AHC 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.

Users taking advantage of this expanded feature also have access to the debugging facility of the system. Currently this means that the TENEX DB 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.

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.

Some non-programming personnel at AHC, 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.

To make this powerful tool more generally usable, an initial documentation of a subset of the LL0 language has been created. This "LL0 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.

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

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

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 INLS) or Jump to Item (in UHLS).

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 viewspec. 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.

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

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 TNI5, with any of the Output to Printer commands, and with the Jump command in UNLS. 

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.

Sequence generator

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

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 indentation; 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

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 L1O 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 LIO program.

User-written LIO 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

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 LIO programs. It may even call other user-written sequence generators.

Online Team Environment
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 LIO User Programs

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

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 LIO Primer.

Examples of simple content analysis patterns and LIO analyzer-formatter user programs follow.

Examples of Simple Content Analysis Patterns

\( \text{BEFORE (25-71-72 12:00)}; \)

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

\( \text{ID} = \text{HGL OR ID} = \text{MFA}; \)

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

\( \text{L 2SLD / \"CA\" / \"Content Analyzer\"/}; \)

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 online team environment.
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 ' = hd/;

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 904-1234.

Examples of Analyzer-Formatter Programs

The following are examples of user analyzer-formatter programs which selectively edit statements in an ALS 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 TXT POINTER sf, pae, eae;
  (outname)PROCEDURE;
  IF FIND sf S+ ' ( +af [']) +ae THEN
    BEGIN
      ST sf + pae S+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) te;
  IF FIND SINCE (25-JAN-72 12:00) THEN
  BEGIN
    SI 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)

introduction

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 1960 when we developed MOLYN, a special, higher-level language, and applied it to all of our NLS programming. MOLYN 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.

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.

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

With the change from our XDS-940 to the PDP-10, we upgraded our compiler compiler to a more flexible Tree-Meta Compiler, and our system-procramming language to the more powerful, less machine-dependent LLO; both developments added to the SEAS tool kit.

We adopted new standards for documentation, and developed several system-measurement sub-systems, see--.743).

During the last year, we developed a source-code debugging system for LLO, working from NLS see--.4120.

Source level debugging not only will be useful to us for the remaining period of our LLO 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 PLI, COENL, FORTRAN, or even an assembly language.

During the last year, we also began development work on the next stage of compiler compiler, and an advanced, modular, system-programming language (XPS) see--.4121), 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.

Source Level Debugging

By taking minor changes to the Texa Dynamic Debugging Technique system, DDT, and to the ARC LLO programming language compiler, and by providing a fairly simple debugging subroutine accessible through NLS, NLS-COM, ARC software engineers have provided themselves with a primitive but effective source level debugging and (procedural level) incremental compilation system.
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,).

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

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.

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

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

System Measurement

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.

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.

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

measuring the elapsed time between two instructions.

Online Team Environment
This is the crudest measurement facility; the only user interface is through the PDP-10's DUT 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.

Measuring the time required by various types of NLS commands.

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.

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.

Monitor measurements

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.

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

Source code documentation standards

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

Standards for documentation and coding were proposed in (Journal, 8573), (Journal, 8637), and (Journal, 8613). They have been used in cleaning up several NLS source code files. This clean up is continuing.

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

Languages

Introduction

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).

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

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.

L10

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.

The June 1971 report (8277,) describes the process of transferring from the XDS-940 languages and compilers to the PDP-10. An 110 Primer (9214,) 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.

**Tree-Meta**

Tree Meta is a metacomplier system for context-free languages developed at ARC. The parsing statements of the metalinguage 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.

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.

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.

In the first 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 VPL compiler.

A version of Tree Meta was discussed in an appendix to the June Report of April 1969 (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.
Modular Programming

Goals

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:

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

(b) they are necessarily constructed from a number of separately developed programs;

(c) they evolve and grow throughout their lifetimes (and there is evidence that they also "age" (1041)).

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.

Desirable Characteristics

Points (a), (b), (c) 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.

We first add two more axioms to the above set:

(a1) Large software systems must be able to take advantage of available hardware for efficiency.
(a5) Program bugs are not known before they occur.

(a6a) All-ah imply that software components, hereafter 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.

(a6) In an interactive programming environment, users must be able to develop and use debugging tools applicable to programs in the same programming system (6039, 10h76).

(a7a, a5, and a6 then imply that

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

(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.

(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.

(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.

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

Online Team Environment 94
To satisfy these objectives, MPS has concentrated on providing the following capabilities:

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.

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

Data definition facilities that:

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.

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.

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.

Online Team Environment
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

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.

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.

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.

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

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.

The EMC has documented its meetings through journal entries as they occurred.

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.

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

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.

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

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

LINAC is to carry out activities within the framework that move us toward the goals, with more detailed, shorter-range plan formulation.
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. These are described in more detail below and in documents (10331), (10034), and (0651), respectively.

FRAMAC

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

FRAMAC's goals and general method of approach are:

- 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, LINC 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.

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

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

a) Piecing together and bring about a general understanding of Dr. Enric Bart'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).

b) Brining 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.

Our Second Stage will include:

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.

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.

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.

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 AKG 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 POLAC, will have to be decided as we progress.

About the initial composition of our FRAMAC group:

We had been visualizing a small FRAMAC group, considering the type of dialog we hope for. But when we reviewed our FRAMAC 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.

LINAC

we have launched an activity within ARC called our Line Activity (LINAC).

LINAC serves several basic needs:

- Modularizing our way of doing things -- something that the size and complexity of our activities require.
- 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.
- Establishing the activity framework within which we can pursue our new-contract commitments to AAPA (as per our proposal of 29 July 1971 -- 7404).

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.

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.

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

Some of our activities will be funded by what amounts to taxation upon all or some of the projects. Such
taxation measures will be established and monitored with due representation by the concerned parties.

ARPA planning and task activities are currently conducted in the following LINAC organization:

**OPERATIONS**

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

**DEVELOPMENT THRUSTS**

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

**PROJECTS**

- **ARPA/MAUC Project: Team Augmentation Portion**
  - Administration

- **ARPA/PADC Project: Network Information Center Portion** (NIC)
  - Administration
  - Computer Service Operations
  - People Service Operations
  - Net Interface (Station Agent and Net Participation)
  - NIC Development

- **ARPA/PADC Project: Mini-Console**
  - Administration
  - System Development

- **ARPA/PADC Project: KPS Cooperation (Xerox)**
  - Administration
  - Modular Programming System Development (MPS)

- **ONR Project: System Developer's Intelligence System (SDIS)**
  - Administration

---

**Online Team Environment**

101
SRI ARC 6 JUNE 1972 13041
Team Augmentation
Internal Organization

**FADC Project: Baseline Management System Development Support Administration**

**SRI OVERHEAD ACTIVITIES**

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

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

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

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

- 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.

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

**POLAC**

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

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 organization and as individuals, to understand how we are doing, and how we can improve.

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.

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.

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

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

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

Each group meets weekly for two hours.

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

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.

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.

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.

It is a way to keep everyone informed about the problems and opportunities facing ARC and its people and its goals.
PODAC has been active for three months at the end of this contract period. It is not yet easy to evaluate our accomplishments.

Meetings of the groups described varied considerably in content.

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.

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.

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

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.

REFERENCES


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

(7670.) Network Information Center, Augmentation Research Center,
Stanford Research Institute, Menlo Park, California 94025. TNIS
Separately paged.

Augmentation Research Center, Network Information Center,
Stanford Research Institute, Menlo Park, California 94025. 1

(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.

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

(8277.) L. C. Engelbart (SRI-ARC). Network Information Center and
Augmentation Research Center, Stanford Research Institute, Menlo
Park, California 94025. Rome Air Development Center, ARPA.

(8661.) L. C. Engelbart (SRI-ARC). To Launch PODAC. Augmentation
Research Center, Stanford Research Institute, Menlo Park,

(9241.) Harvey G. Lentman (SRI-ARC). DEX-2 Proposed Design,
Augmentation Research Center, Stanford Research Institute, Menlo

(9249.) Augmentation Research Center, Stanford Research
Institute, Menlo Park, California 94025. L-10 Programming Guide
(a user guide). 1 April 1972. 130p.

Development of a Multidisplay, Time-Shared Computer Facility and
Computer-Augmented Management-System Research. Augmentation
Research Center, Stanford Research Institute, Menlo Park,
SRI-ARC 8 JUNE 1972

Team Augmentation

References


(10031,) D.C. Engelbart (SRI-ARC), To Launch LINAC. Augmentation Research Center, Stanford Research Institute, Menlo Park, California 94025. 7 April 1972. 66D.

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

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

(10U57,) James C. Norton (SRI-ARC), Initial FRAMAC Meeting Notes. Augmentation Research Center, Stanford Research Institute, Menlo Park, California 94025. 23 May 1972. 1 kp.

(10U59,) 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. 3P.


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

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.

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.

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

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.

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.

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.

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
Introduction

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:

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

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

THE NIC PUBLIC

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

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.

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.

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

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

Present NIC Services

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

Online:

(1) Access to the typewriter version (TNLS) of the augmenting Research Center's Online System (NLS) for communiqué 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.

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

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

(b) Documents are now distributed by:

Online Team Environment
Network Information Center Development and Operation
Present NIC Services

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

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

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

(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.

(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.

(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.

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

(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.

(b) links to other files created by sites with information of potential network-wide interest.

Offline:

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

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

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

Online Team Environment
112
(c) A Station Collection containing a subcollection of documents of interest to Network participants.

(2) Techniques for gathering, producing and maintaining NIC functional Documents such as:
   (a) Current Catalog of the NIC Collection
   (b) ARPA Network Resource Notebook
   (c) Directory of Network Participants
   (d) NIC User Guide

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

(4) General Network referral and handling of document requests

(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.

(6) Crude selective distribution to Station Collections.

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

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

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

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.

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.

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.

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.

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 SKI, 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.

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, alone 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.

OPERATIONS

Computer Service Operations (CSO)

Online Team Environment
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 (#a). A number of changes which appeared necessary as a result of these measurements were made and others are under study.

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 CSN network interface and a new DEC 802 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 62K words of core has been added recently.

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.

The reliability improvements resulting from these measures and others under study should begin to be manifest in the summer of 1972.

Along with these 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.

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 TENEX Protocols are obtained as part of TENEX support from ARPA. 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 (protocol).

People Services Operations (PSO)

During the past year ARC has developed several service functions that are now becoming operational for ARC users and NIC clientele.

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.

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:

1. To increase throughput to meet existing demands.
2. To become capable of expanding rapidly (in throughput quantity) to meet fluctuating service demands.
3. To work at minimizing costs while maximizing responsiveness to customer's needs and values.

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.

Therefore in order to create such a balanced system we launched a new approach to ARC's "people services operations". (see -- 7831,1a)
The main thrusts were:

Organization
Physical Location and Configuration
procedure establishment and documentation
Transcription Activities
Terminals
Personnel
Training

Organization

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.

Physical Location and Configuration

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.

Procedure establishment and Documentation

procedures were devised and documented for:

Use of TNIS (see 7470,) and DEX (see 9934,).

The handling of transcription and other service requests.

All related NIC activities -- clerical and secretarial.

Transcription Activities

Types of work to be handled:

Handwritten drafts
Tape recordings
Dictation notes
Offline documents
Online documents to be edited
Techniques for transcribing material into online files were developed:

Deferred Execution (DEX) covered at greater length see\(^\text{5e212a}\)

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. \(^\text{5e212a1}\)

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. \(^\text{5e212a2}\)

TNLS

TNLS is used largely for routine editing of online documents, and for entering high-priority items during off-peak hours. \(^\text{5e212b1}\)

DNLS

Display NLS is used for difficult editing of online documents and for some highly formatted documents. \(^\text{5e212c1}\)

Receiving processes

We set up a central receiving station. \(^\text{5e213a}\)

One person, with an alternate, handles users' questions regarding job status, time and cost estimates, etc. \(^\text{5e213b}\)

Priority determination process

A requestor specifies his preference for priority:

Immediate service (1-4 hours)
Normal service (4-12 hours)
Deferred service (a week or two)

Online Team Environment

114
Temporary storage of unassigned work

A log system using appropriate work request forms has been set up.

We have a central storage place, organized for control of work by priority.

Assignment process for transcription work

A work scheduler assigns incoming work to group members, balancing priority request with members' capabilities and workload.

Later, priorities may be established by a bidding scheme.

We plan to enlarge this effort to allow assignment to an outside pool of workers trained in DEX, both SRI people and contract manpower.

Output processes

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.

We have developed procedures for delivery of completed work to the requestor.

Terminals

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.

Personnel

We have added several new staff members with contributions to make to VIC. 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.
Training

Classes in TNL5 and DEX were held for ARC and network people. Manuals were prepared. A more detailed discussion of training is given later, see (5e21).

A detailed list of the types of tasks this PSC 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.

PLANNING AND SCHEDULING

Goal setting
Service design
Site Station aid planning
Functional document design
NIC Collection design
Station collection plan
Reference service design
Catalog design
Procedure establishment
Discussion
Procedure writing
Experimentation
NIC facility design
Work flow scheduling
NIC time and cost studies

GENERAL SUPPORT

Dictation
Phone
Orders and financial records
Timecards
Visitor arrangements
NIC travel arrangements
NIC facility upkeep

STATION PHONE ACTIVITY

Station phone answer
NIC outgoing calls

Online Team Environment

121
MAIL SINGLE NIC PIECES

Incoming mail processing
Single mailings

ACQUISITION OF NETWORK INFORMATION

Network resources
Network personnel
Network publication references

CHOICE, ANALYSIS, OF INFORMATION

Analysis for bulletins
Analysis for functional documents
Selection of publications
Abstracting

ACQUISITION OF PUBLICATIONS

Checking holdings
Order form preparation
Receipt, record changing

OFFLINE CATALOGING WORK

Coding
Checking of coding, revision
Proofing and revision
Recoding of old material
Catalog offline records
Old catalog offline work

FILE-BUILDING ONLINE

Input of new citations
Input of old citations
Editing of new citations
Filing of old citations
Bulletin creation
Bulletin editing
Catalog creation
Catalog editing
Catalog file manipulation
Functional documents input

Online Team Environment
122
Let us now look at each of the services provided and see what has been involved in making them available beyond the changes.
Network Information Center
Development and Operations

Operations described above, why they were made available, and some future plans.

There are two major areas of changes to AKC 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.

ONLINE SERVICES

ACCESS TO NLS

The ARC ONLINE System (NLS) is an evolving system which we view as an integrated set of tools for doing general intellectual work (adi). 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.

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.

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 (adi3a).

In thinking about the problems which could exist in supporting all the varieties of typewriter terminals on the network, we (to keep AKC'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 (TNVT or see--601). This protocol has succeeded in allowing access to TLS from different systems and terminals.
NLS, as it has historically developed, is oriented in its command language design for expert users.

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.

At the present time anyone with access to a typewriter terminal connected to the network and with an entry in our identification file (entries can be made in this file directly by network users) can have access to NLS.

We generally find between 1 and 3 users from the network using NLS 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 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.

A system for allowing access to the display version of NLS (DNLS) from the network using IMIAC 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-TFNFX.

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 alphanumeric displays equipped with keyset and mouse.

Documents are presently created by a user at a keyboard device connected to TNS 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-BMCU, 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.

AIDS TO COLLABORATION

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.

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.

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.

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.

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.

New identifications can be created at the time of submission or at other times by use of the Identification system.

The IDENTs are usually a person's or group's initials. The IDENTs are automatically assingned by the Identification System when a person's or group's name is entered into an identification file by use of the system.

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
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.

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).

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.

SAMPLE MESSAGE SENDING SESSION

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.

#Login SP DOE SP DDD SP 1 CR  
(A user named DOE logs into the system - his ID is DDD)

/JUP 11 ON 2Y14 3-AUG-71 17:11/  

%NLS CR  
(The user accesses the NLS system)

/ID:1 DDD CR  

/device: / met-tty/  
("N" signifies that the user DDD is connected through the network with local echoing)

#execute Journal/ (access Journal system)

/submit message/
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)

[interrogate] 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
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
**execute**  **quit**/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** CA  (the user logs out)

EXAMPLES OF ONLINE JOURNAL DELIVERY

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)FWW.NLS:372
*Print Branch .11w1
(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:iw)

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:

1) creation of an image of the documents for each receiver on magnetic tape.

2) Transfer of the tape to SRI's computer center for batch printing.

Online Team Environment

129
3) Document printing.  
4) Bursting.  
5) Stapling.  
6) Checking for correct content and addresses.  
7) Mailing.

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.

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. 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.

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.

To uniquely identify hosts and NIC stations associated with the network we worked with the WG to set up a standard identifying syntax and asked each host and NIC station to name themselves according to the established rules seen--.6cl)

This was a small but important step to help establish uniform communication rules useful to different processes of the APPANII experiment such as various NIC services. These host IDENTs are dialog groups. Thus to send an item to everyone at AHC one uses the IDENT "SPI-AHC" in the distribution list at the time of Journal submission.

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.
Each dialog group or affiliation is a subcollection name as well, and all items sent to that group are automatically part of that subcollection.

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.

Users can find dialog items of interest by use of the catalog listings and indices.

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.

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.

NLS has mechanisms which allow one to "point" at a link and have the system fetch and display the item referenced.

Thus, using links, networks of related documents and dialog items can be created.

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.

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).

Plans to implement this "backlink" capability are being made. This facility will add considerable power and a citation chaining and indexing capability.

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.

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.

Once this problem is solved, further generalization to other non-PDP-10 hosts can be made.

ONLINE ACCESS TO STANDARD VIG DOCUMENTS

Access to VIG 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-node query capability for these users.

Many interactive, online query systems exist which allow one to specify a query by certain keywords or phrases, and logical combinations of these.

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.

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.
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.

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.

The standard NIC documents available online are:

1) TNLS, Journal, Identification, and Number System User documentation
2) Some workbooks for aiding people in learning TNLS
3) The ARPA Network Resource Notebook, describing facilities available at each site which are offered to the network community
4) Catalog of Listings and Indices to the NIC collection of network dialog and network related documents
5) Current Directory of Network Participants
6) Soon the Current Network Protocols document will also be available online

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.
SAMPLE SESSION using <nic>locator

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=8D,5,Light+Slanted>

Load File <nic>locator!

*Print Branch .2lxbn! (Printout of an appropriate view of the NIC documents)

2 NIC DOCUMENTS

2A NIC TNLS USER GUIDE pages=112
2B NIC TNLS EXERCISE FILES pages=23
2C CURRENT CATALOG OF THE NIC COLLECTION pages=404
2D CURRENT DIRECTORY OF ARPA NETWORK PARTICIPANTS
   pages=133
2E ARPA NETWORK RESOURCES NOTEBOOK  Pages=62
2G CURRENT NETWORK PROTOCOLS (not yet implemented online)
2H FOIKLORE...day to day information on NLS  pages=11

*Print Branch .24lebl! (Selection of the catalog and printout with more detail)

2G CURRENT CATALOG OF THE NIC COLLECTION  pages=404:

2G1 INDEX BY AUTHORS  pages=69
2G2 INDEX BY TITLE *ORD pages=333
2G3 RFC LIST BY RFC NUMBER  pages=17
2G4 NIC INDEX BY NIC NUMBER  pages=121

Selection of the author index.

* .2G1 t! (Search for documents by Watson)

**Print Group .watson1 .watson ti!

1192 (watson) NWG/RFC 289: What We Hope Is an Official
List of Host Names

21 Dec 71 6295

1193 (Watson) * Reply to JDL on Output Device Teletype
   20 Dec 71 8289 Watson

1194 (Watson) Summary of 1971 Activities
   9 Dec 71 8150 Watson

1195 (Watson) NWG/RFC 260: A Draft Set of Host Names
   17 Nov 71 8060 Watson

1196 (Watson) NWG/RFC 276: Revision of the Mail Box Protocol (See Number Listing/)

* ♦  (STOP PRINTING)

* ≡1  (Return to Locator)

<NIC>LOCATOR.NLS:1

*Print Branch .2f11! (Printout contents of Resource Notebook)

2F AMPA NETWORK RESOURCES NOTEBOOK Pages=62

  2F1 INDEX pages=23
  2F2 BBN-TENEX pages=10
  2F3 CASL pages=5
  2F4 CARNEGIF pages=5
  2F5 HARVARD-1 pages=3
  2F6 HARVARD-10 pages=7
  2F7 ILLINOIS pages=5
  2F8 INTPO pages=6
  2F9 LL-67 pages=3
  2F10 LL-TX-2 pages=15
  2F11 MIT-AI pages=7
  2F12 MIT-DMCG pages=7
  2F13 MIT-MULTICS pages=11
  2F14 RAND 6a

* ♦  (STOP PRINTING)

*Print Branch .2f12 ±1! (Selection of the MIT-DMCG entry)
Network Information Center
Development and Operations
Online Services

<LISTEH>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
5 V. Interests and Capabilities
6 VI. Login
7 VII. Computer Operator
8 VIII. Miscellaneous
9 IX. Programs
10 X. Programs

<NIC> HIT=DMCC,NLS;2 (Feedback from system as to destination)

*Print Branch .71W1 (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.

OFFLINE SERVICES

Introduction

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.

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.

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.

This concept of a master collection and an associated set
Network Information Center
Development and Operations

Offline Services

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:

1. We need to provide more and better training to Station Agents on how to handle the satellite collections.

2. We need to evolve our cataloging and catalog production tools to the point where stations can maintain their own subcollections and shelf lists.

3. We need to provide selective dissemination of documents to stations based on interest profiles of users at that site.

4. We need to investigate production and use of microfilm technology.

Functional Documents and Their Revision

INTRODUCTION

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 DSN manuals are referred to by NIC as "functional documents.

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.

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.

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.

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.

In preparing a document which is expected to be revised, Network participants are urged to use a looseleaf format.

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.

PROCEDURES FOR REVISION MATERIAL

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.

Inclusion of an additional document in a functional document:

If the added document has already been distributed separately, the holder of the functional document may

Online Read Environment
139
sometimes be asked to insert the old copy in the
functional document. Usually a new document or copy is
supplied.

substantial revision of a bound document, or of more than a
few pages of a loose leaf document:

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.,

NIC 5772
supersedes NIC 5621

Few pages inserted or revised in a looseleaf or
corner-stapled document:

Each new or revised page bears the original document
number, with a date of revision, e.g.,

NIC 5712
3-OCT-71

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.

Deleted pages are replaced by a single page indicating
the deletion, e.g.,

Pages 7-12 deleted, 25-MAR-71

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.

DISTRIBUTION AND TRANSMITTAL PROCEDURES

The transmittal letter accompanying a set of revision
material and the revision material itself constitutes a
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.

**PLANS**

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.

**BUILDING A NETWORK REFERENCE BIBLIOGRAPHIC AND DIALOG DATA BASE**

**COMPUTER-PRODUCED CATALOGS AND INDICES**

**Introduction**

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.

The CSS is concerned with the following principal processes:

- Input, editing, processing, and verification of catalog entries.
Updating of the Master Catalog and subcollection catalogs.

Production of incremental and cumulative, hardcopy and online indices to various collections.

Overall Design Goals and Elements

The basic goals relevant to providing aids to these processes are:

- Maintaining integrity of the master catalog files with maximum protection from both human and mechanical errors.
- Making possible a smooth flow of input from ARC clerks with good facilities for proofing and correcting all clerical input.
- Removing as much load as possible from the computer system during prime use times through the use of deferred execution techniques.
- 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.

The initially implemented element of the Catalog Support System is the Catalog Production Processor (CPP).

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 lists of various kinds, using the Master Catalog as the ultimate data base.

One objective in the design of the CPP, in fact of the rest 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.

The basic design goals which the CPP must meet are the following:
It should permit flexible specification of the types and frequencies of production of the various catalog indices and listings needed by DDS, NIC, DPCS, etc.

It should function as automatically as possible and with a minimum consumption of ARC personnel and equipment resources.

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.

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.

SELECTION OF ADDITIONS TO THE DATA BASE

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.

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.

Data bases thus added include:

A bibliography prepared for use of the attendees at the January 1971 AFIPS Workshop on the User Interface. The bibliographies and indices processed by ARC programs were published in the volume of Proceedings of the workshop. (see -- 9474.)
An extensive bibliography on networking prepared by
Peggy Karr of MITRE. Each reference cited was obtained
in full-size copy and was coded and entered in the
Master Catalog. (see -- 602c,)
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, AFIP, JOU or NIC respectively in its "22 field."

DESIGN OF DATA ELEMENTS

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.

A continuing effort has been the refinement of a set of
data elements. The requirements are:

Data elements should be adequate to describe all species
of information items which are anticipated to be added
to the collection.

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.

The present list of data elements and guidelines refining
their application is supplied. (see -- 600c,). Future
development will study the appropriateness of using
standard data elements being designed by national and
international committees considering bibliographic data.

ENTRY OF ITEMS INTO THE DATA BASE

is 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.

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.

An example of a catalog-entry statement with typical coded data elements:

(N4623) #1 Howard Frank #2 org #b2 Network Analysis Corporation #h 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 6/20. #d1 15 June 1970 #d3 15 October 1969 - 15 June 1970 #f1 r #f2 o #s1 ARPA #6 DAnC15-70-C-0120 #7 OD30 #8 1523 #w1 6-30-71 #w2 DISCUSS.-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; technological optimization; *z2 NIC *z3 new * 5g3d3a

DESIGN OF CATALOG FORMATS

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 -- 5l45, ) and those used in NAS Panel and AFIPS Workshop meetings.

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.

Examples of the listings and indices now produced are:
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

**Online Team Environment**
tools, people, procedures, and training and that integration and development of such a system is a non-trivial process.

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.

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.

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.

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.

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.

DESIGN PROBLEMS

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.

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.

MACHINE PROBLEMS

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.

For some reason yet undetermined, large, heavily manipulated files have gone bad. A great number of files had to be reconstructed from earlier versions.

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.

Delays were caused by printer malfunction. Sometimes several days were lost because the printout for the reproducible master could not be retained until the printer was cured of some aberrant behavior.

Limitations of the line printer caused some compromises with an ideal design. We have experimented with various formats to achieve clarity if not beauty.

PROGRAM PROBLEMS

The complexity of the present catalog production process of calling files and using programs led to
The continual improvement in analyzer and formatter programs required debugging and close examination of results.

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.

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.

The heretofore unreacned 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.

**INPUT TEXT PROBLEMS**

The most obvious problem, and the most common, is misrelling, at the manual coding stage or at the typewriter input stage or by accident in making editing changes.

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.

The selection of information from the document, in the

Online Team Environment

149
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.

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.

REPRODUCTION PROBLEMS

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.

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.

COLLATION PROBLEMS

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.

CONCLUSION

Familiarity with other centers building machine-produced catalogs, (see References Section 2c)
Network Information Center
Development and Operations
Offline Services

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.

SOME FUTURE PLANS

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. 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.

Plans exist to consider in the next year or two distribution of items to site collections on microfilm.

The Directory of Network Participants

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.
COMPREHENSIVE LIST OF IDENTs:

DF    David Farber (UCI)
DGB   Daniel G. Bobrow (BBN-TENEX)
DHL   Duncan H. Lawrie (ILL) DIA
Dor I. Andrews (SRI-ARC)
DL    Dor Limuti (SRI-ARC)
DLM   Dr. L. Murphy (BBN-TENEX)
DLX   Donal L. McNally (Laneker)
DLS   Duane L. Stone (RADC)
DLS2  Daniel L. Slotnick (ILL)
Dr.  Donald McCracken (CMU)

BRIEF LIST OF AFFILIATIONS:

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
           505 Technology Square
           Cambridge, Massachusetts 02139

CCCTF   Canadian Computer Communications Task Force
           100 Metcalfe Street
           4th Floor
           Ottawa 2, CANADA

Online Team Environment
152
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:

BBN-TENEX
B. B. and N. - TENEX Group
Bolt Beranek and Newman Inc.
50 Moulton Street
Cambridge, Massachusetts 02138

DGB Daniel G. Roborow (617)891-1850 ext 330 + Principal
SCB Stephen C. Butterfield (617)891-1850 ext 419 Investigator
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

Online Team Environment
193
EXTENDED LIST OF GROUPS:

BBN-TEMEX  Dan L. Murphy (DLM)  BBN-TEMEX  (617) 491-1850 ext 351
       Bolt Beranek and Newman Inc.
       Computer Science Division
       50 Moulton Street
       Cambridge, Massachusetts 02138

CASE  Patrick W. Fouk (PWF)  (216) 360-2936
       Case Western Reserve University
       Computing and Information Sciences
       11220 Euclid Avenue
       Cleveland, Ohio 44106

CCA  Richard A. Winter (RAW)  (617) 491-3670
       Computer Corporation of America
       365 Technology Square
       Cambridge, Massachusetts 02139

BRIEF LIST OF INDIVIDUALS:

Murphy, Dan L.  (DLM)  BBN-TEMEX  (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 E.  (JHN)  SKI-ARC  (415) 326-6200 ext 1119
Norton, James C.  (JCN)  SKI-ARC  (415) 326-6200 ext 2124
O'Sullivan, Thomas  (TO)  RAY  (617) 762-6700 ext 2124

Online Team Environment
EXTENDED LIST OF INDIVIDUALS:

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) 825-7377  Naficy
UCLA - Network Measurement Center
Computer Science Department
3732 Boelter Hall
Los Angeles, California 90024

Willis H. 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 group, etc.

ARPANET Network Resource Notebook

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 ARPANET 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.

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.
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.

ARPA Network Current Network Protocols

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.

Network Information Center User Guide

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 NLS 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.

We are presently writing a Primer containing a highly restricted subset of NLS, but which is sufficient for people to create and study documents and use the Journal capabilities.

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.

Support of Offline Network Dialog

At the present time the NIC supports several dialog groups, the main ones being the Network Working Group (NWG) of 50 members and its subgroups, the Speech Understanding Research Group (SURG) of 25 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.

Requests for Network Information

People interested in the ARPA Net, from organizations not directly connected with the network, in 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.

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 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.

TRAINING

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-WAG 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.
These people have returned to their sites to use our system and help others at their site learn to use it as well.

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.

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.

As part of this teaching process a series of "workbooks" which take a person through commonly used TNLS operations has been prepared.

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.
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.

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 outages 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. 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.

It is clear that there is much yet to be learned about handling network protocols in local operating systems and in network hardware development.

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.

CONCLUSION

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.

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.

REFERENCES

(5145) Current Catalog of the NIC Collection. ARPA Network Information Center, Stanford Research Institute, Menlo Park, California. (Current version dated 1 February 1972.) Separately paginated.


(6745) ARP Network Resource Notebook. ARPA Network Information Center, Stanford Research Institute, Menlo Park, California. (Current version dated 22 November 1971.) Separately paginated.

(6808) UCB System NWG/75, UCSD Computer Center. University of California at Santa Barbara, Computer Sciences Department, Santa
References


(7814.) James G. 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. 11 p.

(8014.) Anil Bhusan (MIT-LC), Bob Braden (UCLA-CCN), Eric Harlsen, John Reamer (PAND), Alex McKenzie (Batt-NET), John Melvin (SRI-ARC), Bob Sundberg (HPV), Dick Watson (SRI-ARC), Jin Anite (UCSD). NWG/PFC 270, Revision of the Mail Box Protocol. ARPA Network Information Center,

519


520


521

(8295,) Richard W. Watson (SHI-ARC). 446/KFU 284, 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.

522


523


524

During this period, general network participation other than direct Network Information Center activities has been active.

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.

Protocol Development

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.

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.

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-NMCG.
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 (OLS). 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 190, NIC 7141 and revised in RFC 221, NIC 7612 and RFC 278, 6096. 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.

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 OLS 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 OLS from an IMIAC display. The initial work here was described by Deutsch in "DAC PDP-10--IMIAC Communication System," RFC 100, NIC 7136 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.

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.

We have been experimenting with running the display version of OLS over the network from two sites, RBW and UCLA-ARC. The results are encouraging from RBW, which has a 9600-cad interface to their IMIAC and a resident NIC. The UCLA-ARC system runs its Telnet and its NIC as user programs. They have a 1200-cad connection to their IMIAC, when loaded, response is considerably slower than experienced locally at UCLA or experienced by RBW. When both the UCLA-ARC and SRI-ARC machines are lightly loaded, response at UCLA is barely satisfactory.

Charles Irby has been active in the network graphics area.

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 1972.

Some work has been done by the NIC on considering the requirements for a Network Data Management Protocol that would support management of distributed data. We have maintained contact with the people working in

Online Read Environment 164
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.

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.

NETWORK COORDINATION

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 KAND in April 1972.

REFERENCES


(7135,) Peter Deutsch (PARC). NWG/RFC 190, DEC PPP-10 -- IMLAC Communication System. ARPA Network Information Center, Stanford Research Institute, Menlo Park, California 94025. 13 July 1971. 15p.


ARPA Network Information Center, Stanford Research Institute, Menlo Park, California 94025. 20 July 1971. 6p.


(8056.) Abhay Bhushan (MIT-DMGG), Bob Braden (UCLA-CCN), Eric Harslem, John Heafner (RAND), Alex McKenzie (BBN-NET), John Melvin (SRI-ARC), Bob Sundberg (HARV), Dick Watson (SHI-ARC), Jim White (UCSB). NWG/RFC 276, Revision of the Mail Box Protocol. ARPA Network Information Center, Stanford Research Institute, Menlo Park, California 94025. 17 November 1971. 6p.
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.

FIGURE 23. (Opposite) ARC Computer System.

Present Configuration

The present ARC computer facility configuration is as follows:

Digital Equipment Corporation (DEC) equipment is the heart of our facility, providing the computer, core memory, and mass storage devices (discs, magnetic tape units).

PLP-1C

The KAL0 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.

Memory - 8 KAL0's (2 KEL0's being added in June 1972)

These are ferrite core memories and are used with the KAL0 processor. The memory allows for storage of 37-bit words (36 bits and parity) and has a 1 us KAL0/ME10 cycle time. Each memory box has a storage capacity of 16,384 words.
Memory Interface DF-10

The DF10 Data Channel is a high-speed transfer device (10^6 words/sec). In the ARC configuration, it accomplishes direct data transfer between the (RPO2's) and memory.

Once enabled, data transfers independently of the program in progress, thereby releasing the central processor for other operations.

Disc Packs - RPO2's

Our RPO2's each provide storage for 5,196,000 36-bit words. Average access time is 62.5 ms. The transfer rate is 15 ms/word. The ARC system has four online RPO2's for a total storage of about 20 x 10^6 words. An additional RPO2 is available to back up the disc system.

Disc Pack Controller - RP10

The RP10 provides the interface logic between the DF10 Data Channel and the RPO2 Disc Pack Driver.

LTO tape Units

These are special magnetic tape units used for loading programs into the core memory. They are usually used for bringing up the ARC system.

Max Tape Units and Controller

These units enable mass storage of information onto magnetic tape and are used in performing disc dumps and for file archival processes.

Line Scanner

The DC10 Data Line Scanner provides a timesharing two-way interface between the PDP-10 central processor and a maximum of 64 teletype-like stations. The current configuration handles 24 stations.

Folt, Beranek, and Newman, Inc. (FBN) 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.

Online Team Environment
FIGURE 23  AUGMENTATION RESEARCH CENTER-COMPUTER SYSTEM CONFIGURATION
Faking Box

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.

Interface Message Processor (IMP) and IMP Interface

The IMP is the interface between the AIC 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.

Bryant Drum and Interface

The Bryant drum is a mass storage device with a capacity of 1,066,720 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.

Bryant Disk

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 PDP-2's while further uses for it are being considered.

I/O Control Box

This device is used to extend the PDP-10 I/O bus. It also provides manual control over the peripheral devices it interfaces.

External Core (xcore)

This is a 32K 41-bit memory. It is now used for storage of display and keyboard information and other non-critical information transferred at slower rates.

xcore Multiclexer

This attaches 8 ports to xcore. (The main core HALO/HALC's have 4 ports as part of their structure.)
Ycore Interface Box

This device provides for the proper timing and voltage interface between the PDP-10 memory and the Xcore Multiplexer.

Real Time Clock

This clock provides the reference for all times recorded by the system.

TTY Patch Panel

This connecting panel allows some of the many TTY and modem inputs to be connected to the 24 channels available on the line scanner.

Headphones

There are 8 headphones and modems connected to the system.

Display Controllers, Tasker Display Generators, and Closed Circuit TV

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.

Data Products Line Printer

This device provides for hardcopy printout of user and system files.

Test Box

This unit occupies a part of Xcore and is used as a troubleshooting tool.

Input Devices Controller (IDC)

This equipment handles information from the display consoles (12 keyboards, keysets, and mice) and stores it in Xcore to await processing.

A/D Converter
This converts analog mouse coordinates to digital coordinates to be stored in Xcore via IDC.

Display Consoles (12, plus spares)

These each are composed of:

Mouse
Keyset
Keyboard
Video Displays

Console Patch Panel

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.

Illustration to be used:

ARC PLP-lu System Configuration Layout

Problems we have been facing

ARC Service Problems

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.

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 accessibility and user-response at what we consider to be undesirable levels.

The main cause of such lowered service levels has been problems with our hardware, although some software problems have also been encountered.

We have been concentrating on the various hardware problems that have caused lower-than-desired service levels.

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.
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. 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.

A new HBN 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).

The DEC PDP-10 has had several failures in the past few months. DEC responds quickly and effectively to such occurrences.

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.

We plan to discuss with commercial utilities the possibility of their providing all or part of another machine for the NIO portion of our NLS service in an effort to:

1. become more reliable and
2. to provide more computer service to NIO users.

We would have to secure additional funding for such an arrangement.

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.

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.

We have tried swapping off a single DEC diskpack to see if we
can back up the Bryant equipment satisfactorily with that arrangement.

The number of users surportable is considerably reduced, but service to a limited number (1 or 5 users) is satisfactory.

With the added 32k D/C memory (coming in June 1972), there will be less swapping, so that swapping off the disk packs is expected to support more users than during early trials.

In addition, we plan to develop software that will swap off several of the packs, not just one. We also are contemplating adding another disk pack controller.

If swapping off the disk packs doesn't appear to be the best way to provide backup for the Bryant drum, we may have to add another Bryant drum.

The Tasker display system is now five years old and is requiring an increasing level of maintenance and troubleshooting.

Since it provides the primary display facility to ARC UNLS 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.

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.

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.

The Bryant disk 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 D/C disk packs.

The Bryant disk 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 is not put it in the mainline of our system operation.
SYSTEM SOFTWARE

Imlac Support for DNLS

A program written (by Peter Deutschen of Xerox Palo Alto Research Center -- XPACR) for an IIIIAC 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 XPAE personnel (also phone line), at the Network Measurement Center at UCLA (ARPA NET), and at MBN (ARPANET).

TENEX

In our initial use of HB-N-TENEX the main concern was to just "make it work". In getting TENEX to run on our unique hardware configuration we made many extensive modifications and additions. In the ensuing year and a half of experience with TENEX and its evolution we have learned much, with the responsibility of providing reliable computer resources for the NIC many heretofore overlooked requirements in running our facilities have become considerably more critical.

It has become increasingly important to run as unmodified a version from HB-N's distributed version as possible. Modifications are only made when there is a real user need and HB-N cannot or will not make the appropriate modification. When we do make modifications 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 HB-N of the change in the hope that it will become a part of standard TENEX with a subsequent release.

The following is an overview of some of the changes we have made here at SRI-ARC to HB-N distributed TENEX.

Further details, including implementation details, are available for the asking.

OPERATIONAL PROCEDURE CHANGES

CHANGES TO START-UP AND RESTART PROCEDURES OF THE MONITOR

ETMOOT

We have switched from using TMDOFF (for loading the monitor from DECTAP) to using ETMOOT, a DEC-provided replacement for
THEDUMP that is much easier to use and a better program than
THEDUMP.

Novice startup procedures

we have changed the starting address of the monitor from 100
(which goes immediately to DDT) to SYS001.

Thus the procedure necessary for a novice to bring up the
system is easier and is as follows:

readin DTBOOT
type CH.

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 SYSTE.M.SAV
SYSTE.M.SAV is the default name for loading for DTBOOT

DDT Flushing

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.

The system may be run:

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.

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.

Online Team Environment
175
Systen Startup Procedure

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.

If CHECKDSK does not run successfully, then a message is broadcast to all currently connected users telling them that the disc needs fixing.

We allow a dial-up line access in this case, so that a system programmer can fix the disc from home if necessary.

We made this modification with the primitive inter-job communication described below.

Auto-start-up jobs

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.

Compiling and Loading of a New Monitor

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 J RSTs and CALLs to the new code.

We have also broken the MON assembly into swappable and resident code similar to the SWPMON assembly.

Thus we can add code that is logically related to code in the MON assembly but not resident.

We have made several changes in the compile-and-load sequence.

These changes give us more information at each step in putting together a new monitor.

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.

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, IPARAMS, 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:

we have removed the bounds checking that used to be done by going into DDI.

A new program was written that is assembled with and called by POSTLR.

This program does the checking that used to be done in DDI 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 POSTLR.
POSIXL 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 go into MDIT and determine which source files were used to generate this monitor.

PRIMITIVE INTER-JOB COMMUNICATION

We have implemented a very primitive inter-job communication facility.

It involves a system-wide cell with each bit independent of the other bits.

Each bit is directly settable, resettable, and testable.

A process must know the password for any bit to set, reset, or test it.

ADVISE

We have implemented an advise facility similar to that of the 94U.

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.

If lines are input linked, then characters are placed into the big buffer with the line number of the advisee.

HANDLING JUP DISPLAYS

(see also PFC 190 (7135,) and PFC 191 (7136,))

Online Team Environment
we have made many changes to the teletype routines to accommodate our displays.

Basically, we defined an escape sequence:

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.

There exists a mapping from the special sequence to normal TTY input.

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.

This escape sequence enables us to support other types of displays (including IMACS over the NET) with no change to either NLS or TENEX as long as these "remote" displays input the proper escape sequence.

we have made the necessary changes to the rest of TENEX to accommodate this sequence, e.g. STI, and added additional jsys to be able to define what type of terminal (TTY, local display, remote display) is associated with each line.

FAST TERMINAL HANDLING

we have added a jsys to say that padding (sending additional rubouts) is required for this terminal when a CR or LF is output.

This means that the user will not lose the characters at the left margin on fast terminals.

(we understand that version 1.29 takes care of padding. we will get rid of any inconsistencies in our code.)

SCHEDULER CHANGES
we have changed several scheduler parameters to get the kind of response we want.

Primarily, we have tuned the system to give very good service to highly interactive processes and very poor service to more computebound processes. We tuned it by adjusting $QBASE$, $TBASE$, and $TFACTH$.

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.

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 $M$N idea of $I/N$ service to all users. We give our display users a larger share of the machine.

Quite a bit of code has been added at ARC to gather statistics. This measurement code is part of a subsystem (SUPERWATCH) written at ARC to get a profile of the system performance at any time.

Documentation on this system is available (see below).

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.

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 $SUH$, $LSH$, $AOS$ sequence does the job.

The sampling of memory (user pages) is done by a process clock on 500ms intervals. It gives us a profile of memory use.

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.

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.

It indicates the kind of service being provided to interactive users.

If the response index goes over a threshold, the EXLC prevents new logins.

Since our scheduler has been biased against compute-bound jobs, we added a JSYS to set some scheduler parameters (TSAE, TFAITR). This allows us to dynamically alter the scheduling characteristics of our system.

A special subsystem allows the operator to set the parameters to "compile time" or "normal".

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.

MISCELLANEOUS CHANGES

GTJFN

If a version number of -A is specified to GTJFN then the following happens:

If the file exists then the user is returned a JFN for the highest version number of the file.

If the file does not exist then a file is created for the user and he is returned a JFN for this new file.

DELNS:

This is a new jsys that we have added.

Online Team Environment
1d1
Abstract

superwatch is an information gathering and formatting program designed to help find out what is going on within our timesharing system.

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.

Introduction

The system monitoring is done in several steps:

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.

The difference between two meter readings, and the time interval between the readings, can be used to compute an average rate over the interval.

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.

At the end of the collection period, the file is printed in whatever form the user desires.

Monitor meters

The TIMES system, as it came to us, contained several meters, but we found them inadequate in answering our questions about the system.

We added several types of information collection to the system.

We added meters to the monitor, especially with respect to how time was spent in scheduling and other system overhead functions.
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.

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).

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.

The User Program - Superwatch Subsystem

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.

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.

The primary parameter specified by the user is the interval between collections or samples. The program simply dismisses itself for the specified interval between samples.

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.

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.

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.
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.

Generally, the user specifies a set of parameters he wishes to see. The simplest format is a list of the values of each requested parameter at each interval. An average over the entire test is included.

Fault record and PC results are printed in table form giving address ranges, counts and percentages for each range.

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.

A special command allows a real time display (histograms) of specified parameters in real time. This is actually a collection command rather than a prin 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.

Typical use

we usually use the subsystem in one of several ways:

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.

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.

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.
The real time display mode is useful for finding out what is happening when the system is behaving strangely.

There are several parameters which we have found to be very useful:

It is essential to know where the CPU time is going. We deal in terms of percent of real CPU time:

idle time

time spent running user programs

time scheduling

time spent waiting on drum and/or disc

time in system overhead (e.g. network, garbage collection, etc.)

Disc and Drum behavior and usage:

percent of time busy

queue lengths

time to transfer a page, including queue wait time

number of reads, writes

Memory utilization:

number of jobs holding space in memory

amount of memory reserved for above jobs

actual number of pages held by above jobs

number of free pages

number of pages retained due to sharing

usage by subsystem

percent of real time used
compute time between page faults

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.

A Few Discoveries

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.

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.

The P5 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 P5 sampler has been a guide for reorganizing the code in MLS, in order to group frequently used code to reduce the working set size.

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 NFT user to maintain it.

one of our subsystems takes 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 setting 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.

occasionally, we have problems with one 16K memory box. The
standard procedure is to run without it, with 10K 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.

Generally, an information gathering system like ours is valuable:

1. To verify that the system is working as designed.
2. To identify the cause of poor service at the time it is happening (e.g., a bug, hardware malfunction, or just overloading).
3. To identify the "weak link" in the system configuration (drum, disc, memory or CPU capacity).
4. To evaluate changes in the system or hardware configuration.

REFERENCES IV - Computer Facility

(7135,) Peter Deutsch (PARC). NUG/WFC 190, DED PDP-10 -- IMILAC Communication System. ARPA Network Information Center, Stanford Research Institute, Menlo Park, California 94025. 13 July 1971. 150.


Plans
Goals

Online Team Environment
188
Plans

Goals

GOALS

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:

Advancing the techniques available to ARC and Network system builders and users for augmenting the development future Plans of computer-based information systems.

Making the Network Information Center into both

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

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

Moving useful augmentation techniques and services out into the ARPA-Network Community.

In the discussion that follows, and in our proposal to NAOC/ARPA (74NL), we outline the types of activity that seem to us best to meet these goals.

SERVICE TO USERS

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:

Our planned NIC services involve a steadily expanding set of explicit "reference and dialog support" services (sec. -- 74U6). 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.

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.

In general, we enjoy this snow 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).

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 non-dissipative way, with individuals or groups

(a) whom we can adequately support with computer and personnel resources, and

(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.

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

(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.

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

(a) Service Delivery--where these utilities can deliver responsive, interactive transactions, over a complex repertoire of service functions, with both a high degree of reliability and a high degree of availability, and

(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, out with a resource-utilization framework that is balanced between efficiency and demand capacity.
Therefore, we plan to concentrate our efforts within a four-pronged project wherein coordinated advances can be made in:

1. Developing service functions that will be of maximal value in our above-mentioned goal structure,

2. Developing the knowhow and capability for delivering significantly useful service to the Network, as a utility,

3. Developing the knowhow and capability for marketing a utility service to the Network,

and wherein we become ever better at

4. Operating a utility service,

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.

**BASIC PROJECT WORK**

We are planning that under our new base contract, AKC's "utility" would initially serve two, bulk-commodity customers--AKC workers and NIC customers.

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."

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."

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 AKC's internal domain in readiness for marketing them when the time is right.
we will hope to develop a market for our services that leans strongly
toward-ô customers interested in bootstrapping.  

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.  

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).  

Our Base Project work will focus on:  

1) Developing Service Functions for:
   (a) external Users (via the Network)

NIC reference and dialog support functions (discussed further
in -- 7.06,)

Our planned major points of emphasis are as follows:  

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.  

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.  

Reorganize our hardware and software systems to enable
smooth expansion as the need arises.  

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.  

As new dialog support functions are developed and tested
on the research side of the house, move them into
operation in the NIC.

Online Team Environment
192
Provide improved querying capabilities for the online reference files such as the:

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.

Continue to improve making information available by preparing weekly notices of new additions to the NIC collection.

Prepare specialized bibliographies for subjects of wide interest.

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:

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.

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.

(b) Internal Users (evolving toward external use), such as:

Prototype Dialog Support System
(discussed further in -- 7407,)

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, LPCS.

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.

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.

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
Plans

Basic Project Work

be found in a 7 Dec 1969 memo.
(see -- 5220,5e)

Examples of functions under development or being considered are:

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 MLS files of entry details.

Prototype Documentation Production and Control System (DPCS)
(discussed further in -- 7408, )

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.

If we need more throughput to shake down the system, and/or if through VIC'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

Online Team Environment
195
activity would serve as a test bed for the successive stages of PPC system developments we want to go through. 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. We may coordinate this system with parallel use of a high quality CCM 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. The earliest form of a PPCS would have the authors working offline to a large extent, using Deferred Execution (DEX) processes.

The features provided by DEX will allow clerical people to follow complex mark-up notations made by authors as drafts are developed. Spooling-input typewriters, using magnetic tape recording equipment would be used. Subsequent evolution of a PPCS 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.

Prototype Software-Engineering Augmentation System (discussed further in (7809,) and -- 7ull, ) New or changed features being considered are:

Source level debugging and incremental compilation A primitive system is being implemented now with a more advanced version to come following MPL implementation.

Documentation aids
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

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.

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.

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 B&K and so forth.

Further develop our compiler system

Allow easy generation of interpreters as well as compilers.

Modularize compilers.

Possibly generalize trees to networks in Tieve Meta.

Develop program verification capabilities

Development of heuristic programs to determine that programs do what the writer has indicated they do. Allow statement of programmer assumptions at various points in a program. The verification facilities might test those assumptions and specify if they can be false.
Plans
Basic project work

Develop trace facility in the baseline program

to help improve estimation and give a history of our activity.

Develop catalogs and indices for system files

Index procedures by function.

Once a remote site has established a DNS 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 DNS 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.

with straightforward utilization of our compiler-compiler techniques operable through DNS, they can easily build special-purpose languages that match to other computers, to other purposes, at binary or assembly-language levels.

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.

More Advanced Use of the Software Tools at ARC

We have described above how the software engineer at another location might use DNS for writing his programs.

It will also be possible for the remote programmer to use other software augmentation tools developed here.

For instance, the TRADIC compiler writing system could be modified to produce code for another machine (this was in fact done as part of the transfer of DNS from the XDS-9k 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 [7][1].

The compiler-compiler will be composed of modules, so that the code production can be more easily replaced without requiring 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.

In addition, it will open up new ways for the remote programmer to access and use the other tools at ARC.

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

We will design the next stage Handbook, including the specification of content categories together with techniques and procedures for maintenance of the Handbook.

We plan to implement this design during the next contract period.

Indices and Tables of Contents for the Handbook are planned to be designed and implemented.

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 obsolete documents as appropriate.

We expect the Handbook system to aid in stimulating the production of documents that are needed, but missing from our information base.

Online Tear Environment

199
Prototype Baseline Record System
(discussed further in -- 710,)

We plan to improve the data collection procedures and storage mechanisms of the present Baseline Record System.

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 APC data handling systems.

AHC users will be more effectively oriented toward the need for and trained in a more organized task definition and selection process.

Better views of the Baseline Record will be produced for use by AHC, with more useful user-created view capabilities provided.

Hardcopy and Online Baseline Records will be more complete and made a part of the daily working life of AHC researchers.

We plan to develop better methods for keeping the Record up-to-date, both online and in hardcopy.

The ARC resource accounting system, as it develops, will be integrated with and used by the Baseline Record System.

(2) Developing Service-Delivery Principles and Practices for:

(a) Computer services, including considerations such as:
- Remote LNLS
- Remote Hardcopy Delivery
- Reliability
- Resource allocation, accounting, billing
- The questions of scale, efficiency, reliability
- Service-capacity expansion plan

(b) NIC-service (information, people help)

(c) Transcription services

(d) Documentation services (as operational prototype of DPC5, within AHC).
(3) Developing Service-Marketing Principles and Practices:

(a) Learning how to negotiate with prospective customers for delivering various kinds of service to them, including questions such as:

On what basis are the agreements made?  

How is financing accomplished?  

How is the accounting performed?  

How are the scheduling and billing of service delivery accomplished?  

How are conflicts resolved (market conventions, arbitration)?  

What guarantees can practically be made regarding accessibility, reliability, documentation accuracy and completeness, and the like?  

How are user training and helping provided?

(b) Providing Operational Marketing and Delivering of Services:

Developing the framework, as the marketing and delivery systems begin to take shape, in which the current service resources are marketed within the AAR and NIC customer market.  

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.

REFERENCES


Online Team Environment

201
SHI-ARC 6 JUNE 1972 13041

Plans

References


Online Team Environment

202
GLOSSARY

APH -- Acronym for the Arithmetic Processor of the PDP-10.

ARC -- Acronym for Augmentation Research Center.

ARPA -- See DARPA

Arithmetic Processor -- The central processing unit of the PDP-10.

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.

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.

BRS -- Acronym for Baseline Record System

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.

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.

Branch -- In the NLS hierarchy of statements, a statement and all substatements that depend on it.

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.

Center -- The same as ARC.

Compiler -- A computer language that is used to translate from one set of symbols to another, particularly to machine language.

Console -- As used here, specifically a user's control console for the Online Team Environment
AHC's Online System (NLS). The consoles presently in use at AHC consist of a display screen, a keyboard, a "mouse", and a "keyset".

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.

Current Statement Pointer -- The internal symbol fixed on the current statement by NLS.


DDT -- Acronym for Dynamic Debugging Tool, a program useful for establishing at what point in another program a problem occurred.

DEC -- Acronym for Digital Equipment Corporation, the manufacturer of the center's PDP-10 computer and PPD-2 disc memory.

DSS -- Acronym for Dialog Support System

Dialog Support System (DSS) -- The system of files, programs, and procedures at AHC for storing, sorting, and recovering the interchange of thoughts, plans, memos, technical documents, etc. that accompany our system development.

Display Start Statement -- The same as "current statement"

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.

FRAMAC -- From Framework Activity. An organized activity among members of the Center who are involved in planning to define long and short term goals.

Field Operations -- In programming NLS, manipulations that involve the capacity of the PDP-10's software to handle parts of words.

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.
Glossary

graphics, or any combination of these. Conceptually, a file corresponds roughly to a hard-copy document.

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.

HLP -- Acronym for Higher Level Processes

Handbook -- A complete reference work of all systems and activity at the Center at a given time.

Higher Level Processes -- A phrase once used for what we now call User Programs.

IMLAC -- The manufacturer of a display console used experimentally with NLS.

IMP -- Acronym for Interface Message Processors. Hardware devices that code and decode messages for transmission between the computers on the AkPA Network.

Ident -- A two-to-six-letter code, given to people or groups for recognition by the Journal Ident system.

Intellect -- The human competence to make, sort, exchange, and apply knowledge to decision making.

Journal -- The open-ended information storage and retrieval system that forms the core of the Dialog Support System.

JSys -- lit.: "jump to system" The machine instruction used in Tenex to invoke a monitor supplied service; i.e. a jump to a subroutine.

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.

L-10 -- The ALGOL-like language in which our online system is written.

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.

LINAC -- From Line Activity. The line management structure of the
Center, a matrix of projects and functional organization, as
differentiated from PODAC and FRAMAC.

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 NLS commands without regard to what file the user has
loaded. Links have the form (ddu,ff,n;x) where the field ddu contains
a TENEX directory name; the field ff contains a TENEX file name; the
field n contains and NLS statement name or number and the filed x
contains NLS viewspecs. Fields are frequently left to default in
practice.

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.

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.

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.

Monitor -- A program which remains in memory at all times and controls
the coming and going of data and other programs in the machine.

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.

NGG -- Acronym for Network Graphics Group

NIC -- Acronym for Network Information Center, one of ARC's key roles in
the APRA Computer Network. The NIC is a computer-assisted reference
and communication service for information pertaining to the Network.

NLS -- Acronym for the ARC Online System.

NWG -- Acronym for Network Working Group

Network Working Group -- A group of users of the APRA Network organized
to develop Network functions.

Online System -- This is ARC's principal and central computer-based
development in the area of computer aids to the human intellect. As present 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.

Output Processor -- The subsystem of the portrayal generator that processes bit files into sequential files suited to drive devices that produce hard copy.

PDP-10 -- The computer used at the Center from the winter of 1972 until the present. The asynchronous arithmetic Processor has a 1-microsecond cycle and uses 36-bit words based on a 68k page box into 512-word pages.

POD -- Within PODAC, a group of about A employees of the Center that meets weekly for purposes of personal and organizational development.

PODAC -- Acronym for the continuing, organized personal and organizational development activities within the Center.

plex -- In the MLS hierarchy, the set of all statements that have a common source.

Pointer -- An old name for marker.

Portrayal generator -- The class of MLS code that creates something formatted for view by a human.

Protocol -- Among users of the ADA Computer Network, a document describing conventions for carrying out some activity over the network.

PADI -- Acronym for Home Air Development Center.

RFQ -- Acronym for Request for Comments

Request for Comments -- A series of memoranda between Network Liaison personnel numbered and distributed at the Network Information Center. They are no longer restricted to requests for comments.

Ski -- Acronym for Stanford Research Institute.

STIL -- Acronym for statement identifier. A number unique to each statement in a file and that remains with the data regardless of document structure change.
Glossary

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 view-decs.

Server Telnet -- See Telnet

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. 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".

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.

Supervisor -- A group of programs that measures the loads on different pieces of hardware and on subsystems of TENEX and NLS.

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.

TNIS -- Acronym for Typewriter Online System. The system used at ACO 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.

TODAS -- Acronym for Typewriter Oriented Documentation Aid System. The version of NLS used from typewriter-like terminals prior to 1971.

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.

Textpointer -- In NLS, as used on the PDP-10, the fixation of NLS on a space between two characters which allows the users to be sure editing will begin with the following character.

Tree Meta -- The ACO compiler-compiler system, used to compile all the languages at ACO.
User Programs -- Processes in which the basic user features of our online systems (particularly MLS and TNLS) are used as building-blocks in the construction of programs for carrying out specific, perhaps rather complicated tasks.

User Telnet -- See Telnet

Viewspecs -- A feature of MLS 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.

XDS -- Xerox Data Systems, manufacturer of the XDS-940 Computer, used at the center until January of 1971.
Appendices
Handbook Contents
APPENDICES

APPENDIX I CONTENTS OF ARC HANDBOOK

Section 1. USER DOCUMENTS

Volume I NLS GENERAL

NLS CURRENT STATUS
NEW NLS FILES FROM THE PIT

Current Folklore Branch of NLS Status
with new System Changes

AN OCTOBER FEST Including Goto Program

Late July NLS

7/7 NLS

Mid-June Changes in NLS

Files
Proposal for New File Command
Load Locked File Command

ADDRESSING
Link Delimiter Change

EDITING
VIEWS
Views specs, a Brief Table
Views specs

ERROR MESSAGES
Note Concerning NLS Error Messages

Section 1. USER DOCUMENTS

Volume II NLS SPECIFIC

GENERAL

Online Team Environment
NIC TNLS User Guide

VOLUME III
DNLS SPECIFIC

Multiple Display Areas in DNLS

User Features of NLS and TOLAS
(no CATNUM)

Introductory Notes
Keyset and Viewspecs
NLS Commands
NLS Vector Package
Viewchange System
Links and Returns
NLS Content Analyzer
Keyword Information-Retrieval System
Miscellaneous Useful Information
Definitions
Calculator Package

Section 1. USER DOCUMENTS

Volume IV  TENEX

Response to Login Message Change

Another Proposal for Handling Old Versions of NLS Files

Further Currents About Only Keeping One Version of NLS Files Around

EXEC 1.32.Ot FEATURES

EXEC LOGON Command to start NLS automatically:

auto-logout of inactive jobs

Online Team Environment

212
EXECUTIVE MANUAL - PDP-10
(no CATNUM)
Introduction
Executive Language Structure
System Access
TENEX File System
Device Handling
Subsystem Control
Program Control and Debugging
Queries
Terminal Characteristics Commands
Index

SNDMS - A new Subsystem

TLINK - A new Subsystem

Know Your Disc Space

Link/Advise Refuse Default

Insert from Tecc File or Whatever Specs

Proposal for a Simple Archive System and
Directory Size Limitation

Directory Trimming Program

Comments on Dump and Bad Files

TENEX Operating System and EXEC

TENEX User's Guide
- Not online

Section 1. USER DOCUMENTS

Volume V DIALOG SUPPORT SYSTEN

GENERAL
VIO Journal System User Guide

JOURNAL SYSTEM
Proposed New Journal Submission Features -
Pre-Specified Catalog Data

Online Team Environment
Appendices

Handbook Contents

Journal System

A Suggested Procedure for Journal Operations 7637
Journal Error Recovery Guide 6402
Online Journal Delivery via INITIAL File 7291
Proposal for Changing Journal Command Structure 6961
What to do about Had Journal Numbers 7822
Proposed Modification to the Place Link syntax in the Journal 7624

ID SYSTEM
Response to ID system changes proposal -- 8680 7800
Proposed Minor Changes to the Identification System 8730
Followup on 6680 and 6730 -- Identification System Changes 8680
Identification system 9143
Syntax and Semantics of TALS Identification Submode 7590
ID System Changes 7346

NUMBER SYSTEM
Number System 7639
What to Do About Had Journal Numbers 7824

DISTRIBUTION
Notice of removal of hard copy for SKI-ARC members 8721
Author Option to Refuse Hardcopy 6607
Distribution Techniques 8074
Quickie Instructions for Hard Copy Distribution 6025
Hard Copy Distribution Operator’s Guide 6219

Online Tear Environment 214
| Changes to Journal System (Hard Copy Distribution) | 6216 |
| MISCELLANEOUS | |
| Journal System Command Summary | 7640 |
| Individual Idents | 7641 |
| Group Idents | 7642 |
| Affiliation Idents | 7643 |
| Index to NIC Journal System User Guide | 7644 |

Section 1. USER DOCUMENTS

Volume VI

| OUTPUT PROCESSOR | |
| Output Processor Directives | 7477 |
| Output Processor Directives, Notice of a Tutorial File | 8542 |
| Output Processor Brief User Guide | 6912 |
| Output Processor Reference Guide | 6978 |

Deferrred Execution System

Deferred Execution User Guide (in progress)

Section 1. USER DOCUMENTS

Volume VII

| L10 | |
| Programming Guide (in progress) | 7052 |
| L10 Documentation-Formal Description |  |
| Design for Generalization of User Programs | 7527 |

SEQUENCE GENERATOR

How to Make Your Own Sequence Generator Programs 6013

Online Team Environment 215
## Appendixes

### Handbook Contents

#### CONTENT ANALYZER

| Content Analysis Language                        | 8420 |
| Design for Generalization of User Programs       | 7527 |

#### SORT MERGE

### Section 2. SYSTEM DOCUMENTS

<table>
<thead>
<tr>
<th>Volume I</th>
<th>CATNUM</th>
</tr>
</thead>
<tbody>
<tr>
<td>NLS</td>
<td></td>
</tr>
<tr>
<td>Second Proposal for NLS Command Language</td>
<td>9351</td>
</tr>
<tr>
<td>WSC and SUBSTITUTE</td>
<td>9452</td>
</tr>
<tr>
<td>New Versions of NLS</td>
<td>8661</td>
</tr>
<tr>
<td>Bugs and Bug Discoveries About 89k6</td>
<td>8951</td>
</tr>
<tr>
<td>(Jump Command glitch)</td>
<td></td>
</tr>
<tr>
<td>Proposed Control Language for NLSDDT</td>
<td>9198</td>
</tr>
<tr>
<td>atldj - new jsys</td>
<td>8782</td>
</tr>
<tr>
<td>Copying HEL-NLS and NLS Files to NIC-NLS</td>
<td>8663</td>
</tr>
<tr>
<td>Cleaning NLS -- reply to WD's 8637</td>
<td>8637</td>
</tr>
<tr>
<td>More on Clean NLS...Call for meeting on Error handling</td>
<td></td>
</tr>
<tr>
<td>More new versions of NLS</td>
<td>8914</td>
</tr>
<tr>
<td>Bug: Jump Command Doesn't XeArm Cursor</td>
<td>8940</td>
</tr>
<tr>
<td>Comments on Cleaning up NLS</td>
<td>8573</td>
</tr>
<tr>
<td>Comments on Cleaning up NLS</td>
<td>8637</td>
</tr>
<tr>
<td>Response to NLS Clean-Up Suggestions</td>
<td>8581</td>
</tr>
<tr>
<td>Suggestions for Cleaning NLS</td>
<td>8573</td>
</tr>
<tr>
<td>Toward a More Consistent Command Language in NLS</td>
<td>8179</td>
</tr>
<tr>
<td>DDT-Resistant NLS Bugs</td>
<td>8519</td>
</tr>
</tbody>
</table>

---

**Online Team Environment**

216
Handbook Contents

NLS Utility Background Processor Description/Users Guide  7371

Current Usage of Program Communication Flags in TENEX by NLS  7849

New NLS Highcore Arrangement  8397

New Printer Driver  8172

Use of SIGNAL Construct for NLS Error Machinery  6209

110

on parameters in

comments on 110  9267

primitive Text Macro  9265

possible Explanation of DDT-Resistant Bug  9254

Request for Change in NLS to use new JSYS  8523

110 Documentation  8098

TREF META

JOURNAL

Program for determining proper startup of Journal Background Process  7052

Comment on starting Journal Background Manually  8794

A Proposed New Format for Journal Transmission Notifications  8930

Descriptive Notes about DSS, a Dialog Support System  8368

Proposed Journal Changes  7272

Acknowledgement of IN System Changes  8405

Proposed New Journal Entry Mechanism  6508

Proposed Journal Changes and the New File System  6369

Online Team Environment  217
SRI-ARC 8 June 1972 13041
Appendices
Handbook Contents

| New Note on Journal                   | 7026 |
| Journal Error Recovery                | 7291 |
| Response to On-Line Journal Distribution Note | 6962 |
| Some Journal Changes                  | 6348 |
| Hard Copy Distribution Operator's Guide | 6219 |
| More Notes on Hard Copy Distribution  | 6347 |
| Communication Flag Usage              | 7372 |
| A Note on Revised Slinker Startup Procedure | 7370 |

**ID SYSTEM**

| New groups in the IDENT system        | 9248 |
| New Identification System Program: LMEMLIST | 8924 |
| More on Ident System Mods...in response to 5143  | 9114 |
| Affiliates and Groups in the Identification File | 7462 |
| Get Field Routines for Identification System | 7525 |
| Signatures in the Context of the Identification System | 7657 |
| son of Group/Affiliate in Ident File    | 7489 |

Section 2. SYSTEM DOCUMENTS

| Volume II                              | CATNUM |
|                                       |       |
| BASELINE                               |       |
| A suggestion for facilitating the updating of Baseline Information (in progress) | 7634 |
| The Next step in ARC Baseline Planning |       |
| SORT MERGE                             |       |
| Sort here, and Update Primitives      | 7660 |
| CATALOG                               |       |
| Catalog Taking Problems: File size and where is AA | 9306 |

Online Team Environment
216
# Appendices Handbook Contents

- Request for Policy Determination re: Handling Seasonal Dates in the Catalog System 8944
- Catalog Support System Implementation Plan 8005
- Catalog Support System Design Proposal 8004
- Catalog Maker's Diary 8299
- Catalog Making Problems Fall '71 8308
- NLS Catalog Producing Programs and Catalogs 7402

**FILE SYSTEM**

- Initial Outline for MPS Activity Plan 6259
- proposed Journal Changes and the New File System 6425
- "User Implementation for Stage 0 File System (Preliminary)" 6920
- Initial Spec for NLS/User-File System Interface Routines 6256

**MODULAR PROGRAMMING SYSTEM**

- what I have been doing since July 9153
- MPS Progress since last July: a personal view 9114
- Change in MFL object file 8931
- Recent MFL/MPS changes 5700
- A System for Modular Programming 7053
- The Modular Programming System: Processes and Ports 7357

**DEBUGGER**

- Status Communiqué on NLS Debugger 9312
- Proposal for Control Language for Primitive Source Level Debugger (Updates 8162) 6334
- Extension to Proposed Debugger 6177

---

Online Team Environment 219
# Appendices

## Handbook Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Parser for the Primitive Source Level Debugger</td>
<td>8162</td>
</tr>
<tr>
<td>proposal for Primitive Debugging, Terminal Linking, and DNLS Complex Commands</td>
<td>8161</td>
</tr>
<tr>
<td>Primitive Debugging Sys Proposal w/New DDT Implement Details</td>
<td>8100</td>
</tr>
<tr>
<td>DEX</td>
<td></td>
</tr>
<tr>
<td>DEX-2 Design my best thoughts</td>
<td>9319</td>
</tr>
<tr>
<td>DEX-1 Initialization--Case Shifts from TI's with Cassettes permitted</td>
<td>9133</td>
</tr>
<tr>
<td>DEX-2 Proposed Design</td>
<td>9241</td>
</tr>
<tr>
<td>26 January DEX-2 DESIGN MEETING NOTES</td>
<td>8734</td>
</tr>
<tr>
<td>proposal for Sequential File Input Subsystem for DEX</td>
<td>8605</td>
</tr>
<tr>
<td>DEX-II Design Revised as per 10-Jan Meeting</td>
<td>8567</td>
</tr>
<tr>
<td>SUPERWATCH</td>
<td></td>
</tr>
<tr>
<td>Known performance problems</td>
<td>9313</td>
</tr>
<tr>
<td>NEW SUPERWATCH COMMANDS</td>
<td>9303</td>
</tr>
<tr>
<td>Another View of what Superwatch is Saying</td>
<td>8765</td>
</tr>
<tr>
<td>the system as seen by superwatch, etc</td>
<td>8761</td>
</tr>
<tr>
<td>all about superwatch</td>
<td>8649</td>
</tr>
<tr>
<td>Preliminary Superwatch Documentation</td>
<td>8544</td>
</tr>
<tr>
<td>MISC</td>
<td></td>
</tr>
<tr>
<td>OMIV is faster now</td>
<td>9316</td>
</tr>
<tr>
<td>program for making sequential (NLS) file from directory listing</td>
<td>8957</td>
</tr>
<tr>
<td>Integrating the reformatted Ident file into MLC Locator</td>
<td>9145</td>
</tr>
<tr>
<td>Risk Pack - Drum Simulation Analysis</td>
<td>8228</td>
</tr>
</tbody>
</table>

Online Team Environment

220
Funfil February 24, 1971

DE: NLSCC JSY 8098

Note on NLSCC JSYS

SYNLAC Update

Results of Disk-Pack Study

Description of HELP and TLNEXLX Procedures and Use

The State-Changing Problem for Remote Computers

Experience with the Network

TLNEX Configuration Guide

Section 2. SYSTEM DOCUMENTS

Volume III - TLNEX

GENERAL

Reply to ASN about EXEC 1.32.03

Some Comments on New EXEC

EXEC 1.32.03

The TLNEX Scheduler

Proposal for Changing TLNEX Monitor Source Files

JSYS

MTMCI JSYS -- HOW TO COUNT NETWORK TTY USERS

NEW JSYS - STRM

NEW JSYS - STRAP

NEW JSYS - SET-m

REQUEST FOR CHANGE IN NLS TO USE NEW JSYS

NOTE ON NLSCC JSYS

Online Team Environment
appendices

Handbook Contents

Re: NLSCR JSYS

TENEX JSYS Manual - A Manual of TENEX Monitor Calls

UTILITIES
- program communication flags
- reclaiming resident dxt
- disc space allocation/restriction in tenex
- system disc file allocation
- update to disc allocation/restriction writeup
  - 6758 (See Number Listing)
- ttypmod
- Alternative algorithm for printer queueing
- Lineprinter and Spooling
- updates flushing updates about dxt flushing
- more comments about dxt online printer stacking
- new Printer Driver

ARC/TENEX
- sri-arc changes to standard tenex
- current usage of Program Communication
  Flags in TENEX by NLS
- reply to 6717 on Old Versions of NLS Files
- more on old versions of NLS files

ADMINISTRATIVE
- Results of MENTR FC and NET use
- Identfile Problem (RBN-NET)
- new TENEX Group Assignments

Online Tea: Environment
222
NEW TENEX DIRECTORY GROUP

NOTES ON CHANGE TO CONFIGURATION

GROUP

HUGS
PARAMETERIZING HUG CHECKS AND HITS

TENEX HUGS FROM SHI-AI

125 HUGS

MISCELLANEOUS

A REQUEST FOR COMMENTS ABOUT STARTUP SWITCHES

CENTRONICS PRINTER

SOFT DRUM FROP PRINT-OUT SPECS

"SAKE OF TENKA PROGRAM COMMUNICATION FLAGS"

UPDATE REGARDING DDT FLUSHING

DATA TERMINAL IN SUPPORTING DOCUMENT

REASONS WE DON'T USE HLS FOR MONITOR FILES

NEW HANDLING OF RESIDENT DDT

SCHOOL PROBLEM

MAIL BULLETIN LINKING

SECTION 3: PROPOSALS

VOLUME I

NETWORK INFORMATION CENTER AND COMPUTER TEAM

INTERACTION - PART ONE - TECHNICAL PROPOSAL

MARCH 1, 1972

PREPARED FOR HATIE AIR DEVELOPMENT CENTER (ISIK),

GRITFIS AIR FORCE BASE, HAM, NEW YORK 13400

TECHNICAL SUPPORT FOR MADC USE OF AUGMENTATION

ONLINE TEAM ENVIRONMENT
Handbook Contents

Technology - Part One -- Technical Proposal
Prepared for: Rome Air Development Center (ISIM),
Griffiss Air Force Base, Rome, New York 13440

Development of a Small Computer-Augmented
Information System - Part I -- Technical Proposal
December 20, 1971
Prepared for: Information Systems Branch,
Office of Naval Research, Department of the Navy,
Arlington, Virginia 22217

Network Information Center and Augmentation System
Development - Part One -- Technical Proposal
July 29, 1971
Prepared for Rome Air Development Center (ISIM),
Griffiss Air Force Base, Rome, New York 13440

APPENDIX A: An Introduction to the Structure
and Evolution of NLS

APPENDIX B: NLS Development - Function and
Operational Delivery

APPENDIX C: Dialog Support System

APPENDIX D: Document Production and
Control System

APPENDIX E: Software Engineering Augmentation
System

Experimental Development of a Small
Computer-Augmented Information System
January 25, 1971
Prepared for Information Systems Branch,
Office of Naval Research, Department of the Navy,
Arlington, Virginia 22217

Section 1. REPORTS

Volume I - 1967

Special Considerations of the Individual As a
User, Generator, and Retriever of Information

Online Team Environment

244
April 1961
Prepared for: Annual Meeting of American Documentation Institute, Berkeley, California

A Augmenting Human Intellect: A Conceptual Framework 3900
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 Research,
Washington 25, D.C.

Augmenting Human Intellect: Experiments, Concepts, and Possibilities 9641
March 1963
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

Computer-Aided Display Control 9692
July 1965
Prepared for: National Aeronautics and Space Administration, Langley Research Center, Langley Air Force Base, Virginia

Section I. REPORTS

Volume II 1967-1968

Display-Selection Techniques for Text Manipulation 9691
March 1967
Prepared for: IEEE Transactions on Human Factors

Online Team Environment 225
Appendices

Handbook Contents

In Electronics, Volume III, Number 1

Study for the Development of Human Intellect Augmentation Techniques
March 1967
Prepared for: National Aeronautics and Space Administration, Langley Research Center, Mail Stop 126, Langley Station, Langley, Virginia 23365

COPE: An Assembler and On-Line-CRT Debugging System For The CDC 3160
March 1968
Prepared for: National Aeronautics and Space Administration, Langley Research Center, Mail Stop 126, Langley Station, Langley, Virginia 23365

MOLQUL: Preliminary Specification For An Algol-like Machine-Oriented Language for the SIS 940
March 1966
Prepared for: National Aeronautics and Space Administration, Langley Research Center, Mail Stop 126, Langley Station, Langley, Virginia 23365

Development of a Multidisplay, Time-Shared Computer Facility and Computer-Augmented Management-System Research
April 1968
Prepared for: Rome Air Development Center, Griffiss Air Force Base, New York 13441

MOLQUL: A Machine-Oriented ALGOL-Like Language For the SIS 940
April 1968
Prepared for: National Aeronautics and Space Administration, Langley Research Center, Mail Stop 126, Langley Station, Langley, Virginia 23365

Study for the Development of Computer-Augmented Management Techniques
18 October 1966

A Research Center for Augmenting Human Intellect
Fall 1966

Online Team Environment

226

Section II. REPORTS

Volume III 1969-1970

Study For the Development of Human Intellect Augmentation Techniques
January 1969
Prepared for: National Aeronautics and Space Administration, Langley Research Center, Mail Stop 126, Langley Station Hampton, Virginia 23665

Study For the Development of Human Intellect Augmentation Techniques
16 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
6 March 1969

Study For The Development of Human Intellect Augmentation Techniques
16 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)
November 21, 1969
Prepared for: Rome Air Development Center, Griffiss Air Force Base, Rome, New York 13441

Computer-Augmented Management-System Research and Development of Augmentation Facility
April 1970

Online Team Environment

227

Intellectual Implications of Multi-Access Computer Networks
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

Section L, REPORTS

Volume IV 1970-1972

Advanced Intellect-Augmentation Techniques
July 1970
Prepared for: National Aeronautics and Space Administration, Langley Research Center, Langley Station, Mail Stop 126, Hampton, Virginia 23665

Network Information Center and Computer Augmented Team Interaction
18 May 1970

1970 ARL ACTIVITY SUMMARY
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)
QUARTERLY MANAGEMENT REPORT 5 (Covering the period 9 February 1971 through 8 May 1971)
QUARTERLY MANAGEMENT REPORT 6 (Covering the period

Online Team Environment 22d
9 May 1971 through 8 August 1971

QUARTERLY MANAGEMENT REPORT 7 (Covering the period
9 August 1971 through 8 November 1971)

Experimental Development of a Small Computer-Augmented
Information System
April 15, 1971
Prepared for: Information Systems Branch,
Office of Naval Research, Department of the Navy,
Arlington, Virginia 22217

Experimental Development of a Small Computer-Augmented
Information System
April 15, 1972
Prepared for: Information Systems Branch,
Office of Naval Research, Department of the Navy,
Arlington, Virginia 22217

Section 5. THINKPIECES

Volume I
brief notes on software meeting to discuss
ARC expansion -- 11-NOV-71

Notes on ARC Demonstration Techniques

Ideas Concerning ARC Technical Seminars

Some Thoughts on PODAC

ON GETTING FROM HERE TO WHERE?

Thoughts Deriving From the XEROX Research Proposal

Some Questions for ARC

Terminal response time

To launch PODAC

Intellectual Implications of Multi-Access
Computer Networks

Note on future saletype services from NIC and PINS,

Online Tear Environment

224
Appendices

Handbook Contents

and accounting-system implications. 7606

Library Automation with Distributed Resource Sharing via Computer Networking 7323

Some NP Notes on a Bootstrap Community 7310

Some Miscellaneous Leave-Behind Notes 7311

Notes on Matters of ARC Organization 7308

Notes on Possibility of ARC Giving System Support to Other Sites' Documentation 7306

Rough Notes on Possibility of AFC Giving System Support to Ames ILLIAC Documentation 7294

Descriptive Notes about USS, a Dialog Support System 7272

ARC/IPT Project-Continuation Th1nkpiece 7271

NP Note about Journal-entry process 7016

Discussion Log: DGK with PJL on DEX-1 design 6996

Notes, Planning ARC Internal 6934

Network Graphics Meeting Notes 7463

Transcription of discussion on features in PDP-10 TODAS, 1 February 7060

Proposal for Changes to the ARC TENEX File Group Write Access Configuration 7016

Section 6. PROCEDURES

Volume I

BASELINE

Proposal for Changes to the Baseline Record System 8064

Proposed Scenario for the Baseline Record System 8168

Description of Current Baseline Record System 6975

Online Team Environment 230
SYSTEM
New Versions of NLS 8661
A Proposal to Create NLS User and User Feature Groups 7851
Proposal for Journal Logging Teletype 7897
Redesign and maintenance of NLS status 6122
A proposal to establish software teams 8334

NIC
Delivery for the Network Proposal 7363

CATALOG
Master Catalog Entry System Design Proposal 7930

PSO
APC TELEPHONE/MESSAGE SERVICE Proposal 9240
PSO is alive and twitching! 8420
Outline for Establishing People Service Support Team (PSS) at ARC 7834

DOCUMENTATION
Initial Requirements for APC Handbook Document Collection 7630

LIBRARY
Selecting Report Publications from USGHA 7830
ARC New Document Bulletin No. 1 9391

GENERAL
To License PODAC 8651

Section 7. ADMINISTRATIVE

Volume I
AHC
San Comm. Weekend System Usage Schedule -- effective 8-MAR-72 9381

Online Team Environment 231
Appendices

Handbook Contents

Meeting notes -- CIHAD -- 3-DEC-71 --
software maintenance

Tape -- UCE and Staff--PODLAUNCH, 2 February 1972

POD
PODCOM
PODCOM Minutes of 29 February

REdWOOD
REDWOOD POD Minutes March 7,1972

Redwood POD Meeting Notes, 13 March 1972

Redwood POD Notes: 22FEB

Redwood Pod Notes...Feb 22

Redwood Pod Notes...Feb 22

REDWOOD POD - 2  3 FEB '72 Meeting Notes

FIR
Fir POD Minutes for Meeting of 14 March 1972

Fir POD Meeting, 9 Feb 1972

Fir POD Minutes -- 1 Feb

Fir POD Meeting Notes

OAK
CEdAR

cedar agenda

Communique from the Cedar 9, containing two
OFFICIAL SUGGESTIONS

Cedars Arise and Form Your Roots

Communique from the Cedar 9 -- 16 February 1972

Communique from the Cedar 9 -- 9 Feb, 1972

Online Team Environment

232
Communique from the Cedar 9 -- 2 Feb. 1972
Communique from the Cedar 9 -- 26 Jan

EMC

EMC NOTES 6-MARCH-72
EMC 7 Feb and 10 Feb Meeting Notes (JCH)
EMC 3 Feb 72 Meeting Notes
EMC Meeting Notes -- 31-JAN
EMC Meeting Notes 17-JAN to 27-JAN
EMC Meeting Notes 6-JAN, 11-JAN, 13-JAN
EMC Meeting Notes 6-JAN-72
EMC Meeting Notes 4-JAN-72
EMC Meetings Notes, December 22 and 27, 1971
EMC Agenda--December 17, 1971
The EMC Agenda and Notes
EMC Agenda and Notes for 11/2/71
Agenda and Notes for ARC EMC Meetings
11/9  11/11  11/15
More Thoughts on ARC and the EMC
on SHI and ARC Working Hours
Agenda and Notes for ARC EMC Meeting--Tuesday, 9/28
Menu (old) on establishment of ARC's Executive Management Committee
Agenda and Notes for ARC EMC Meeting--Tuesday, 9/21

SOFTWARE
Ident System Meeting

Online Team Environment
Appendices
Handbook Contents

NEW COMPUTATOR FOR NEW NLS SYSTEMS

Notes on Software Group Meeting of 10 January 1972

Tape - Software Meeting #1; November 1971
Tape - Software Meeting #2; December 1971

MISC
MESSAGE TO ALL ARC REGARDING SIGN-OUT FOR TERMINALS.

suggested schedule for the use of the System

some suggestions wrt regard to scheduling computer usage.

XEROX meeting notes -- 15-FEB-72

Tape - Dialog Support System
Tape - PSS7 Meeting with DCE; October 1971

Section - 5. CATALOGS and INDICES

Volume I

NEW ARC JOURNAL Indices Note to ARC (to 28 JUN 72)

ARC JOURNAL INDEX BY NUMBER (PDP-10 entries only)

ARC JOURNAL INDEX BY AUTHOR (PDP-10 entries only)

ARC JOURNAL INDEX BY TITLEWORD (PDP-10 entries only)

Online Leaf Environment
APPENDIX II NIC DOCUMENT LOCATOR

INTRODUCTION

The Locator consists of tables of contents for selected NIC documents.

The tables of contents extend to two or three levels.

Through Locator the user can retrieve useful parts of documents online with a few commands.

The headings in the tables of contents contain links to the corresponding online documents.

The links include viewspecs that curtail print commands so they display table-of-contents-like samples of the object documents.

In the syntax for commands below, CA is control- and the text enclosed in square brackets is echoed on full-duplex terminals.

USING THE LOCATOR ONLINE

To load Locator:

Syntax: S (nic,locator, CA)

To print the introduction and instructions:

Syntax: p/print/ c/ranch/ .1 CA (1, w CA (1)

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.

Syntax: p/print/ c/ranch/ .2 CA w= CA

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.
Appendices

NIC Document Locator

Syntax: print/ b/ranch/ .STATEMENTNUMBER CA xeb CA

To load the corresponding title or subtitle in the document itself, print the branch with an up arrow.

The up arrow searches the statement in locator for a link and then follows it.

The system then prints the branch in the file named in the link.

Syntax: print/ b/ranch/ .STATEMENTNUMBER SP / CA (/\ CA (\)

When you use an address that loads a second file, the system echoes the directory and file name.

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.

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 viewspec set to display the complete text.

Syntax: print/ s/statement/ .STATEMENTNUMBER CA = CA

To return from a file to which you linked to the current file, use the jump to file return command.

Syntax: SF * CA.

The linkstack command will print out a list of the last five files you have loaded.

Online Read Environment

236
Syntax: `execute st/atus l/inkstack`

NIC DOCUMENTS

NIC TNFS USER GUIDE  pages=112

PREFACE  pages=1
(Journal, 7470, 5:MSCT)

SYNTAX CONVENTIONS  pages=2
(Journal, 7470, 6:INDCT)

CONTENTS  pages=2
(Journal, 7470, 7:INDEX)

THE TNFS OPERATING SYSTEM AND EXECUTIVE  pages=13
(Journal, 7471, :X)

FILE STRUCTURE, CONTENT & INPUT/OUTPUT OPERATIONS  pages=15
(Journal, 7472, :X)

ADDRESSES IN THE NLS SYSTEM  pages=13
(Journal, 7473, :X)

CREATING AND VIEWING TEXT  pages=16
(Journal, 7474, :X)

TEXT EDITING  pages=6
(Journal, 7475, :X)

DEVICE CHARACTER SETS  pages=6
(Journal, 7476, :X)

OUTPUT PROCESSOR DIRECTIVES  pages=12
(Journal, 7477, :X)

ERROR MESSAGES  pages=9
(Journal, 7478, :X)

COMMAND SUMMARY  pages=7
(Journal, 7479, :X)

GLOSSARY  pages=5
(Journal, 7480, :X)
INDEX pages=11
(Journal,7661,:x)

NIC JOURNAL USER GUIDE pages=76
CONTENTS pages=5
(Journal,7635,:x)

INTRODUCTION pages=10
(Journal,7636,:x)

THE JOURNAL pages=14
(Journal,7637,:x)

IDENTIFICATION SYSTEM pages=10
(Journal,7638,:x)

NUMBER SYSTEM pages=5
(Journal,7639,:x)

SUMMARY OF JOURNAL SYSTEM COMMANDS pages=5
(Journal,7640,:x)

COMMAND SUMMARY pages=4
(Journal,764c,:x)

INDIVIDUAL IDENTS pages=12
(Journal,7641,:x)

GROUP IDENTS pages=1
(Journal,7642,:x)

AFFILIATION IDENTS pages=3
(Journal,7643,:x)

INDEX pages=6
(Journal,7644,:x)

NIC TNIS EXERCISE FILES pages=23

XED...tutorial file in line editing and structural editing pages=13
(Journal,xed,:x)

APADOCCP...tutorial file in manipulation of partial copies
CURRENT CATALOG OF THE NIC COLLECTION  pages= 404

INDEX BY AUTHORS  pages=69
(nic,authind, entry:w)

INDEX BY TITLE WORD  pages=333
(nic,titleind, xentry: wd)

RFC LIST BY RFC NUMBER  pages=17
(nic/rfcindex, entry:w)

NIC INDEX BY NIC NUMBER  pages=121
(nic,nunindex, entry:w)  pages=119

CURRENT DIRECTORY OF ARPA NETWORK PARTICIPANTS  pages=133

BRIEF DIRECTORY OF AFFILIATIONS  pages=6
(nic,briefaff, entry:wdm)

BRIEF DIRECTORY OF GROUPS (with coordinators)  pages=1
(nic,briefg,entry:wdm)

BRIEF DIRECTORY OF INDIVIDUALS  pages =15
(nic,briefi, entry :wkm)

COMPREHENSIVE LISTING OF IDENTs  pages=30
(nic,complistic, entry:wmn)

DIRECTORY OF ENTERPRISE AND ZENITH NUMBERS  Pages=1
(nic,dirent, entry:wmn)

EXTRAFUL DIRECTORY OF AFFILIATIONS (and members)
pages=20
(nic,xtndaff, entry:wm)

Online Leaf Environment
239
EXTENDED DIRECTORY OF GROUPS (and members) pages=30
(nic,xtnidgr,entry:$Hn)  12c5g

EXTENDED DIRECTORY OF INDIVIDUALS (with addresses) pages
=29 (nic,xtnidid, entry:$on)  12c5h

AKPA NETWORK RESOURCES NOTECOK Pages=62  12c6

INDEX pages=23
(nic,resindex,xentry)  12c6a

OBN-TELEX pages=10
(nic,obn-telex, :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=6
(nic,intro, :x)  12c6h

LL-67 pages=3
(nic,ll-67,1:x)  12c6i

LL-7X-2 pages=15
(nic,ll-7x-2,:x)  12c6j

MIT-AI pages=3
(nic,mit'ai,:x)  12c6k

MIT-DMC5 pages=7
(nic,mit-dmc5,:x)  12c6l
MIT-MULTICS pages=15
(nic.mit-multics,:x)

HAND pages=7
(nic,rand,:x)

SDC pages=9
(nic,scd,:x)

SRI-AI pages=6
(nic,sri-ai,:x)

SRI-AKC (NIC) pages=9
(nic,sri-akc,:x)

SU-IIT pages=1
(nic,stanford,1:1)

UCLA-CCN pages=13
(nic,ucla-ccn,:x)

UCLA-NMC pages=7
(nic,ucla-nmc,:x)

UCSD pages=7
(nic,ucsd,:x)

UTAH pages=6
(nic,utah,:x)

CURRENT NETWORK PROTOCOLS (not yet implemented online)
(folks)...day to day information on NLS pages=11
(documentation,folkslore,:x)

OTHER LOCATORS

Mitre resource-Locator pages = 2
(Mitre-tir,resource-locator,1:ct)
Appendices

Typical Resource Notebook Entry

Online Team Environment.
APPENDIX III A TYPICAL NETWORK RESOURCE NOTEBOOK ENTRY

UCSB Computation Center  
IBM 360/75  
IMP #3  
HOST #0

I. Personnel

Area Code is: 805

A. Administrator:  
charles loepkey 961-2261

B. Software:  
don Stoughton 961-3454

C. Hardware:  
rot Ploger 961-2442

D. NIC Station Agent:  
Connie Rosewall 961-3271

E. NIC Technical Liaison:  
don Stoughton 961-3454

F. Principal Investigator:  
david Harris 961-2531

G. Operations Supervisor:  
steve Neumann 961-2274

Mailing address is:

Computer Center  
University of California at Santa Barbara  
Santa Barbara, California 93106

Online Team Environment

2a3
II. Installation Type

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 MV/3 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.

III. Equipment

A. The computer at this site is an IBM 360/75 with a memory size of 3,524,796 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.

B. Peripheral equipment includes:

1. 1 2440 card read/punch unit (1000 cpm read, 300 cpm punch)
2. 2 1403 line printers (132 columns, 1000 lpm)
3. 16 2314 disc drives (28M bytes each)
4. 2 2415 magnetic tape drives (one 7-track, one 9-track)
5. 1 digital incremental plotter
6. 7$ storage tube remote graphics terminals

IV. Consoles

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.

V. Physical Resources

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 #608 "UCSB SYSTEM 360/75")

<table>
<thead>
<tr>
<th>Day</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mon</td>
<td>5:00 am to 10:00 pm</td>
</tr>
<tr>
<td>Tues</td>
<td>9:00 am to 10:00 pm</td>
</tr>
<tr>
<td>Wed</td>
<td>9:00 am to 10:00 pm</td>
</tr>
<tr>
<td>Thurs</td>
<td>9:00 am to 10:00 pm</td>
</tr>
<tr>
<td>Fri</td>
<td>9:00 am to 10:00 pm</td>
</tr>
<tr>
<td>Sat</td>
<td>12:00 noon to 6:00 pm</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>Day</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mon</td>
<td>6:00 am to 10:00 pm</td>
</tr>
<tr>
<td>Tues</td>
<td>9:00 am to 10:00 pm</td>
</tr>
<tr>
<td>Wed</td>
<td>9:00 am to 10:00 pm</td>
</tr>
<tr>
<td>Thurs</td>
<td>9:00 am to 10:00 pm</td>
</tr>
<tr>
<td>Fri</td>
<td>9:00 am to 10:00 pm</td>
</tr>
<tr>
<td>Sat</td>
<td>12:00 noon to 6:00 pm</td>
</tr>
</tbody>
</table>

In addition, network users of batch may run at other times on an irregular basis. The Computer Center is always open.

B. Initial experimental use of OLS can be conducted under a special user number. The relevant accounting parameters are specified in PFC-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 (MOC #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:

- **Central Processor**: $2.123/minute
- **Core Storage**: 0.0002388/KCS
- **Printer**: 0.7539/1000 lines
- **Card Reader**: 0.92176/1000 cards
- **Card Punch**: 3.07242/1000 cards
- **Channel Interrupts**: 3.16/1000
- **High Priority Service**: 25% surcharge

**Online System**

- **Console Connect Line**: 54.00/hour
- **Core Storage**: 0.1776/KRH
- **Computer Resource Units**: 0.06011663/CRU
- **Disk Storage**: 0.10/KRH

**Online Team Environment**
Typical Resource Notebook Entry

Printer Form

Two part
$0.012/page

Three part
$0.019/page

Four part
$0.025/page

Five part
$0.033/page

Six part
$0.040/page

Cards

(2000/box)
$2.75/box

Labels

(three across, 36 labels/page)
$0.06/page

Disk Packs

Storage
$2.00/month

2316 - rental
14.50/month

Magnetic Tapes

2400 ft. - purchase
$18.00/each

2400 ft. - rental
1.50/month

1200 ft. - purchase
$3.00/each

Online Team Environment

247
Appendices

Typical Resource Notebook Entry

Storage - any size
C.50/month

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 (6095 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.

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.

VI. Interests and Capabilities

The UCSD 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 UCSD's research effort has been directed toward development of its Online System.

VII. Login

Specifications for logging into OLS through the Network are contained in RFC #71 (NIC #5417). The most recent user's manual for OLS is on file with the NIC. Specifications 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).

VIII. Computer Operator

Online Team Environment
Communication with the operator through the network is not possible. The operator may always be reached by phone at (805) 961-2274.

IX. Miscellaneous

None

X. Programs

The only processes presently accessible to network users are OLS and KIT 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.

A. FORTRAN IV (IBM: levels G & H) (University of Waterloo: WATFOR, WATFIV) - a high-level language oriented towards mathematical problems.

B. PL/I (IBM: level F) (Cornell University: PLC) - a high-level, general purpose language.

C. PLOT (UCSS) - a package callable from FORTRAN and PL/I programs which allows display of graphical data on a digital plotter.

D. RPG (IBM) - a language for generation of business-type reports.

E. ASSEMBLER (IBM: levels F & L) (University of Waterloo: level D) - a low-level language for systems programmers.

F. SSP (IBM) - a scientific subroutine package for FORTRAN and PL/I.

G. GPSS (IBM) - a high-level language oriented towards the social sciences.

H. SPSS (Stanford) - a set of statistical routines oriented towards the social sciences.

I. SORT/MERGE (IBM) - a program for sorting and merging data sets.

Online Iean Environment.
Appendices

Typical Resource Notebook Entry

J. BIOMED (UCLA) - a set of statistical routines for FORTRAN users.

K. UTILITIFS (IBM) - a set of programs for the manipulation of data sets.

L. OSWIS-II (University of Michigan) - an organized set of integrated routines for investigation with statistics.

M. COBOL (IBM: level 1) - a high-level language oriented toward business problems.

N. ALGOL (IBM: level F) (Stanford University: ALGOL-W) - a high-level language oriented toward mathematical programs.

O. CSMP (IBM) - a high-level language oriented toward modeling problems.

P. SNOBOL (Bell Labs) - a string manipulation language.

Q. XTAB/FREQ (UCSB) - programs for cross tabulation and frequency count.

Online Test Environment

250
APPENDIX IV NIC FUTURE SOFTWARE PLANS

INTRODUCTION

What follows is a compilation of the APC software tasks that are foreseen as important to the NIC, and hence for which the NIC should assume partial or total responsibility.

NETWORK

MOTIVATION and SOFTWARE REQUIREMENTS:

To exploit the network -- via forthcoming, network-standard protocols -- in delivering the NIC's services to the network community.

SPECIFIC PROJECTS:

Inter-host File Transfers:

To hosts' file systems

MOTIVATION:

To enable remotely-generated text files to be entered into the NIC for:

Manipulation via NLS

Retrieval of files previously archived into the Net

Remote submission of journal entries

To enable the transmission of NIC files to remote hosts for:

Return of NLS-manipulated files

Manipulation by remote text editor

Storage at remote host

Archiving of ARC files
Output on remote hosts' printers
Network Journal delivery

SOFTWARE REQUIREMENTS:

Interface ARC-TEENEX to the Network FTP
Provide a mapping between NLS, tree-structured files and ASCII sequential files
Network text editors should be surveyed to determine the constructs which exist in them, in order to provide a reasonable mapping

Interface ARC-TEENEX to the Network Mailbox Protocol
Interface the Journal via the mapping to the FTP and/or Mailbox Protocol

To CCA's Data Computer

MOTIVATION:

To support archiving of NLC files on the trillion-bit store
To allow private, textual files to be meaningfully transferred between NLC and the Data Computer

SOFTWARE REQUIREMENTS:

Represent NLS' file structure using the constructs of the Data Language and provide the mechanism for transferring files between the two systems
Provide the NLS user with the tools for manipulating Data Computer data bases

TO OTHER NLS

MOTIVATION:
To distribute the NIC computing load over several hosts by enabling

Several instances of NLS to function cooperatively

The processing and front-end (command interpreter) components of an NLS system to reside in different hosts

SOFTWARE REQUIREMENTS:

Develop and implement protocols for:

Common or cooperative:

- Number Systems
- Identification Systems
- Journals
- Catalog searches across host boundaries

To other, Network information-handling systems

Network Graphics

Support NLS use from Network graphics terminals

MOTIVATION:

To support use of DMS from refresh-display terminals in the Net, so that the full power of NLS (compared to TNLS) is made available to Network users of the NIC

SOFTWARE REQUIREMENTS:

Interface NLS to:

- Network-standard graphics protocol

- Specific hosts/terminals (with non-standard protocols) as interest dictates

Online Team Environment
Trouble-shooting is inevitably required at the remote host, for each of the above activities.

NLS

System Development

MOTIVATION:

NIC has an obvious interest in promoting the continued development of NLS, the primary tool which it offers its users.

Although the NIC is presently primarily concerned with those system changes which benefit TNLS (since TNLnet 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.

SOFTWARE REQUIREMENTS:

Take an active part in NLS development, sharing in the software load, and assure implementation responsibility for those features which primarily benefit users of the NIC.

SPECIFIC PROJECTS:

NEW, TNLS-specific features

Novice thru expert modes

MOTIVATION:

To isolate the novice user from advanced concepts which would only confuse him, while providing the sophisticated user with access to the full capabilities of the system.

SOFTWARE REQUIREMENTS:

Stratify TNLS in such a way that levels of capability consistent with the user's expertise can be provided.
The impending, wholesale revision of NLS syntax may be just the vehicle for realizing this need.

Lingering statement numbers

MOTIVATION:

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.

SOFTWARE REQUIREMENTS:

Integrate into NLS, some of the capabilities inherent in DEX.

Line-drawing construction

MOTIVATION and SOFTWARE REQUIREMENTS:

To provide a mechanism by which line drawings can be constructed from a TTY-like terminal for later display in DMS.

New, NLS-specific features

shared display screens

MOTIVATION:

To promote the real-time cooperation of NIC users in the construction and examination of NLS files.

SOFTWARE REQUIREMENTS:

Provide a mode of operation in which a user's display can be replicated on another terminal, and the user positions of each user displayed on both screens.

Graphics

MOTIVATION:

Online Real Environment
Software Plans

To support line drawings in NIC documents

SOFTWARE REQUIREMENTS:

Restore the mixed-text graphics features of the old, 940 system

MOTIVATION:

New features common to TNLS and DNLS

New file constructs

SOFTWARE REQUIREMENTS:

To support such constructs as comments, back links, sets, etc.

MOTIVATION:

New file constructs

SOFTWARE REQUIREMENTS:

A complete re-write of the NLS file system is planned.

GROUP DOCUMENTS

SOFTWARE REQUIREMENTS:

To lend support to activities involving the manipulation of documents by groups of users.

MOTIVATION:

Group documents

SOFTWARE REQUIREMENTS:

Automatically keep track of changes made -- what, when, by whom -- and update indices and tables of contents.

QUERYING TECHNIQUES

SOFTWARE REQUIREMENTS:

To ease for both the user and NIC personnel the task of locating within the NIC database, information about specific subjects.

MOTIVATION:

QUERYING TECHNIQUES

SOFTWARE REQUIREMENTS:

Automatic generation of:
file-global indices, tables of contents, and bibliographies (from links).

Data-base-global subject and subcollection indices

Consider the feasibility of applying a question-answering system as a front-end in the search procedure.

Document-keyed virtual items

MOTIVATION and SOFTWARE REQUIREMENTS:

Permit a catalog number to be used as an item for purposes of Journal distribution, meaning the set of individuals to whom the referenced document was distributed.

Calculator

MOTIVATION:

To support the inclusion in NLS files of tabular, numeric data, and provide convenient means for manipulating such data.

SOFTWARE REQUIREMENTS:

Restore the calculator system which existed on the 910 system.

Major structural changes to VLS

MOTIVATION:

The NIC should assume responsibility for participating in the implementation of major system changes which will improve the performance and/or maintainability of NLS.

SOFTWARE REQUIREMENTS:

Participate in the implementation of:

The proposed new file system, which will

Online Team Environment

257
Software Plans

permit a generalization of NLS' current tree structure.

The Modular Programming System (MPS), which will ease the task of checking out new system components and of monitoring system behavior.

Miscellaneous

MOTIVATION and SOFTWARE REQUIREMENTS:

Implement such features as addressing by content, the saving of viewchange-viewspec information, and whatever additional features prove to be necessary or desirable.

System Maintenance

MOTIVATION and SOFTWARE REQUIREMENTS:

The NIC has an obvious responsibility for and interest in participating in the maintenance of NLS.

This responsibility includes activities which:

- Locate and fix bugs
- Reduce the cost to the user of using the system
- Decrease response time by improving code efficiency
- Improve reliability

SPECIFIC PROJECTS: statistics gathering

MOTIVATION and SOFTWARE REQUIREMENTS:

To obtain information concerning:

- The cost of each NLS command to help locate those points in the software which should be made more efficient.
The frequency with which specific commands are used, and then to evaluate the worth of supporting and maintaining infrequently used commands.

The difference in system overhead between network and local users to evaluate the performance of network-related code in the Monitor.

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.

**Resource allocation control**

**MOTIVATION AND SOFTWARE REQUIREMENTS:**

To implement mechanisms for controlling the allocation of such system resources as CPU time and secondary storage among local and network users.

**Literal collection and feedback by the Monitor**

**MOTIVATION AND SOFTWARE REQUIREMENTS:**

To evaluate the effect upon system performance of moving responsibility for literal collection and feedback from NIS to the Monitor, and if the effect is found to be significant, to implement the change.

**Augmentation of secondary storage**

**MOTIVATION:**

To increase the amount of secondary storage available to house the NIC data-base.

**SOFTWARE REQUIREMENTS:**

In addition to the possibilities already described for archiving files in the Net, to...
support continued development of the backup system, which archives on tape.

**MOTIVATION:**
To decrease the amount of time required to assign a catalog number.

**SOFTWARE REQUIREMENTS:**
Generate each number by a computational process, rather than selecting it from a free list.

**Journal System**
Ease operations

**MOTIVATION and SOFTWARE REQUIREMENTS:**
To improve the mechanics of journal operation so that routine functions can be carried out by an operator, rather than a systems programmer.

Make it possible for the operator to recover from crashes.

**Reduce delay to the user**

**MOTIVATION:**
To reduce the amount of time the user must devote his console to the submission process.

**SOFTWARE REQUIREMENTS:**
Provide a mode of operation in which only interrogation of the user is performed on-line, and all other processing done in the background.

**Reduce the cost to the user**

**MOTIVATION and SOFTWARE REQUIREMENTS:**
Reduce the cost of submitting a Journal article to about $0.25.

Cope with the volume of Journal data

**MOTIVATION:**

To efficiently manage a continually growing collection of data

**SOFTWARE REQUIREMENTS:**

- Provide automatic movement from one Journal directory to the next.
- Provide automatic archiving.
- Integrate Journal and Master catalogs
- Improve effective Output Processor performance

**MOTIVATION:**

To reduce the delay to the user of outputting a file for output on the printer.

**SOFTWARE REQUIREMENTS:**

- Run the Output Processor as an independent fork in parallel with other activity at the terminal, or in the background.
Appendices
TNLS Commands
APPENDIX V  A LIST OF TNLS COMMANDS

A list of currently available TNLS commands follows. They are described in detail in the TNLS User Guide. (See 7470).

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
para
FR80
printer
sequential
Teletype
Compiler
Assembler
Sequential
Print
statement
branch
pley
group
Replace entity
Substitute
Transpost entity
Update File
Viewspea
xset
. " Show point
; " Comment Command
Give context commands
\ " Backslash Command
/ " Slash Command
↑ " Up Arrow Command, jump to back