```
  LINEPR.FAI:                                                      1b

     Insert after LP1OPN:
       IFDEF LPTDVF,<
               LOCK LPTLCK,<PUSHJ P,LCKTST>
               MOVE IOS,LPTSTS                                     1b1

  SWPMON.MAC:                                                      1c

     P.33, was:
       XEFACT: SKIPN ACCIFG
               JRST 0(2)
               BUG(CHK,<EFACT FAILURE>)
     Should be:
       XEFACT: SKIPE ACCIFG
               BUG(CHK,<EFACT FAILURE>)
               JRST 0(2)                                           1c1

Exec 1.38 Bugs:
--------------                                                     2

  X1CMD.MAC:                                                       2a

     P.11, at .CHANG: +7, following:
               STDIR
                CALL SCREWUP
```

```
        Insert another:
                CALL SCREWUP                                         2a1

        P.29, at USERN: +4, was:
                SETZ A,         ;SAYS NO RECOGNITION
                STDIR
        Should be:
                MOVEI A,1       ;SAYS NO RECOGNITION
                STDIR                                                2a2
```

DCE 17-MAY-72 18:22 10536

Possibility for SEAS to Take on a Small Contract with CIRAD

This memo requests specific action from HGL and the SEAS team. It also announces a small commitment by ARC to help set up and run some experiments, using our software people as subjects to perform some information-access tasks under observation. 1

BACKGROUND 2

We have been conducting a dialogue with representatives of CIRAD (and a bit with their sponsor, DR. John Goodenough, Air Force ESD) off and on for some months. They are working on a "Computer Aided Software Maintenance Study." Over most of this dialouge, CHI has been directly involved. 2a

For background documentation about their work see: their proposals to ESD, XDOC items (7588,) and (8622,); their reports, (7585,) and (7586,); some correspondence and notes, (9157,) (9158,) (9159,) (9160,) and (9161,). 2a1

The second stage of their work, which was given support by Goodenough and is now under way, includes plans for some experiments with software-people subjects solving various kinds of information-access problems in source-code and documentation material. 2b

Early in the discussions, I agreed that we would consider giving some help for some experiments that might want to calibrate NLS as an aid to such information-access situations -- I have made it clear all along that our involvement would have to be held to a minimal drain on our people/computer resources (not being a question of what they would be willing to pay for). 2c

Richard Overton and Paul Colen from CIRAD have been their main discussants. Overton arranged for a CIRAD systems programmer, Mrs. LarkIn Veigel, to spend two days here a few months ago to get a bit familiar with our NLS and software environment. This was when Diane Kaye was just learning, and they worked together. 2d

Based upon that experience, Veigel prepared (11 Apr 72) a thinkpiece toward an approach, see XDOC (10060,) and (10061,), which I had to rule against because: 2e

it was too large a project in terms of the probable level of support it would require from us, and 2e1

it din't promise that much experimental return as compared with a more straightforward utlization of the current NLS. 2e2

In the telephone discussion with Overton and Veigel about this thinkpiece (24 Apr 72), I suggested that they try to formulate an approach that didn't require any changes in NLS, nor any training of subjects in our ways of working -- both of which I feel would tend to involve too much of our resources. Even if they planned to do the programming and the training, we would need to spend a lot of time helping to make the progamming project come along, and to make the training an experimenting seem reasonably representative of our set of tools and techniques. 2f

AN APPROACH FOR CIRAD-ARC COLLABORATION 3

Recently (3 May 72), Veigel produced another thinkpiece, XDOC (10062,) and (10063,), which was judged by CHI and me to be suitable for basing the joint CIRAD/ARC development of a trial plan. CIRAD is a bit behind schedule in getting its experiments planned, and would like to follow up with this part of its project as soon as possible -- establishing a plan, setting up whatever CIRAD-to-ARC contract might be necessary to cover our costs, and getting under way. 3a

It is quite logical that our participation in this effort be done under the SEAS activity; so this memo is asking HGL, as acting coordinator of SEAS in WHP's absence, to coordinate a meeting with CIRAD people who are currently planning on visiting us Wednesday, 24 May 72, arriving about 0930 (Mr. Paul Colen and either Dr. Richard Overton or Mrs. Larkin Veigel). I set this date as being the earliest that Harvey could meet (getting back from SJCC), the last date on which CHI could participate (before his vacation, in order to help make a smooth handover), and the soonest that I could count on being back from Washington. (If this date won't work, Harvey can call Mr. Colen, (714) 621-3942) 3b

The purpose of this meeting will be to work out a plan of collaborative action, based upon their thinkpiece (nnnn,), that: 3c

gives CIRAD a reasonable boost toward their study goals, whether or not there is any further collaborative work with us; 3c1

gives us reasonable confidence that it won't require more than a few man weeks from our staff to help in the experiment designs, any training for their staff, any help in monitoring the experiments and collecting data, the time of our staff used as subjects, and help in reviewing and interpreting the results. 3c2 and hopefully, produces data and insights that will be of value to us. (To some extent, the more valuable and relveant this activity promises to be to our SEAS thrust, or to our user studies in general, the more we could afford to get involved.) 3c3

I would like JCN to be on call during CIRAD's visit to help if/when needed on contracting questions. 3d

What has been happening in Oak Pod

To responsed to Doug's request for a record of PODAC I interviewed all the members of Oak POD, except Walter Bass who was out of town, during the week of April 10-14. I recorded the interviews and wrote the following account from the recording on May 9. In the interviews I asked a series of standard questions to make the interviews comparable. I did not always ask the questions in the same order. Diane Kaye interviewed me. I encouraged people to express themselves freely and neglect answering questions if they preferred not to. I promised each interviewee that I would keep the tapes to myself. 1

As I reported elsewhere (Ijournal,1036,,) concensus is not characteristic of our POD, I therefore report question by question, giving for each question a general impression of the answers and some apt quotes rather than to drawing overall conclusions. 2

Note, however,people that seem to have scattered opinions about the design and purposes of PODs but to agree on what has actually happened e.g. answers to the first two questions are diverse, wheras they are pretty much the same on the question of frustration or the content of the meetings. 3

What did Doug have in mind when he formed PODs? 4

Although many people said they did not fully understand and two people simply answered "no", this question produced the longest answers. In general people asserted that he wanted to promote personal and organization development, but it turned out that people had very different notions of what personal and organizational development is. 4a

"The way people develop themselves via organizations and the way organizations develop themselves via people." 4a1

"To replace Fadiman -- to handle personal problems." 4a2

"We'd started to be like a software house ... he wanted to bring us back to the total picture of social organization ... something weird." 4a3

"To help the organization to into an organization in which people do their work better because of the organization rather than do their work in spite of it as is often the case now. 4a4

"An experiment ... for some purpose I can't imagine ... because he hasn't seemed to use what PODs have done." 4a5

"To establish a forum where ARC people could discuss issues of personal and organizational development ... not as a dodge to avoid confronting the software people on some particular issue ... organizations of people need to view themselves as improving in order to be healthy ... which should include some analysis. 4a6

Have they done it (What Doug had in mind)? 5

Even people who said that they didn't know what Doug has in mind often had opinions as to whether it was happening. 5a

For instance, one person who'd answered "no" said, "They might be doing it (what he had in mind),..and Doug doesn't like that." 5a1

The person who remarked about people developing via organizations said "people always say they want freedom and then when you give it to them ... they say, 'What do you want us to do?'" 5a2

"No ... we are still very cautious." 5a3

The person who talked about people needing to view themselves as improving said, "I don't think it has worked out near is expectations." 5a4

Are you frustrated in PODs? 6

We all answered yes except one who said, "no... I go to sleep." 6a

Much of the frustration centered around the amount of time devoted to getting organized and deciding what we should talk about, etc. Some other frustration centered around people who had specific hopes for PODs that had been frustrated or the related process of people vetoing one another's ideas. 6b

"We've been struggling too much with people's varying expectations." 6b1

"I'm frustrated by having a predetermined idea in my mind and seeing such slow progress." 6b2

"There is such a disappointing amount of overhead time if we would only allow ourselves the luxury of doing what we really wanted." 6b3

"I've had some fine conversations walking away from the

POD" ... (they were frustrated) "by the size of the group and the unexpressed rule that you don't interrupt." 6b4

What would you like PODs to do? 7

This question produced the widest variety of answers. 7a

"Deal with personal problems, not trip off into the structure of ARC or SRI or the national budget." 7a1

"Talk about how we feel about how organization things here have gone ... what we think about the relation between ARC and SRI ... the goals of ARC ... learn something about group dynamics ... have more organization ... have a rotating facilitator, with a little bit of an agenda." 7a2

"Bring in people who know about things, experts in running groups, scientology, E.S.P., ..." 7a3

"To be a way for people's fantasties about ARC to accrue reality either by shaping ARC more or working out how they don't." 7a4

"I would make a group that could take some of the responsibility for running the organization, but that's not what Doug wants." 7a5

Are PODs good for anything? 8

Answers to this question ranged from "no" to "of course". In general, they were weakly positive. The good cited tended to have to do with communication. 8a

"I felt some comradship ... I feel there is a place to bring up questions, an audience for certain things." 8a1

"Yes ... broadening communication." 8a2

Will PODs go on indefinitely? 9

Answers ranged from "no" to "yes". In general we felt they would probably go on but with changes. 9a

"Not the way ours is going ... it will either get better or die." 9a1

Has the content of meetings changed? 10

We agreed more on this question than any other. In general we felt that it had changed from one meeting to another but not overall. 10a

"Each meeting has its own character ... but we're still hung up in the same mechanism ... there is no particular progression." 10a1

What's at stake for you personally in PODs? 11

Answers ranged from "nothing" to "my interest in ARC". In general we asserted their our stakes were small and had to do with either a chance at communication or getting to know people. 11a

Has the PODs given you a clearer idea of how ARC operates? 12

In general we answered no, but a few people answered yes. No one answered yes strongly. 12a

"Yes ... I've been disappointed to see how deeply problems are built into ARC ... Doug is not going to change, but I've seen new little ways of getting around them." 12a1

"has ... about the business end." 12a2

Have PODs given you a clearer idea about what you can get out of ARC? 13

Most people answered no here too. It's interesting to note that there seemed to be no correlation between people feeling they got had clarified their ideas about ARC and length of service. 13a

"Yes because I've learned that ARC is pretty flexible, willing to experiment ... that people fill a variety of roles." 13b

Are PODs compulsory? 14

The answers to this question were most evasive. When pressed, five people said yes and three people said no. Note these are different numbers from those that appear in (Ijournal,10236,,). Ijournal,10236 Typical evasions were "by Doug's directive, yes, by our POD, no." 14a

Have PODs helped you in regular work with other members of OAK? 15

There were 2 no's and 7 yes's, one strong yes. Someone commented that not being in other PODs sometimes hindered his work. 15a

Have PODs improved the atmosphere here? 16

The general answer was a moderate yes, but there was a kind of uneasiness about the yes as if the motion had been sideways. 16a

"They diffused a touchy situation (the software group)". 16a1

"The question, 'What are we supposed to be doing here?' has not really been answered, but people feel better about it." 16a2

"Well, they give people more to talk about -- everybody know what a POD is." 16a3

Some remarks seemed to me illuminating but didn't fit neatly under questions, 17

"POD talk isn't very healthy talk because while we're talking about it, we aren't doing it." 17a

"I keep thinking my head must be in the wrong place, there is something I don't understand or accept." 17b

"Somebody should know what's going on." 17c

"There's time to do what everyone wants, meet where everyone wants." 17d

"I'd like to know what other people tell their friends about their job." 17e priority message

this is not,repeatnot, a sample message. however,it is a simple message. 1

ahi

ahi training is complete 1

ahi

(J10541) 18-MAY-72 11:07; Title: Author(s): John L. McNamara/JLM; Distribution: Duane L. Stone, James H. Bair, Thomas F. Lawrence, James C. Norton, Paul Rech, Dirk H. van Nouhuys, Richard W. Watson, Douglas C. Engelbart, John L. McNamara/RBMS RWW DCE JLM; Sub-Collections: RADC RBMS; Clerk: JLM;

revtpol4

ft wants demo 19may 1

revtpol4

(J10542) 18-MAY-72 11:20; Title: Author(s): Thomas J. Barcalow/TJB; Distribution: James C. Norton, Roger B. Panara, Joel P. Cavano, Dirk H. van Nouhuys, John L. McNamara/JCN RBP JPC DVN(how are you) JLM; Sub-Collections: NIC; Clerk: JLM;

Sample Journal Session

This is a sample medsage 1

(J10543) 18-MAY-72 11:22; Title: Author(s): Joel P. Cavano/JPC;
Distribution: Duane L. Stone, James H. Bair, Thomas F. Lawrence, James
C. Norton, Paul Rech, Dirk H. van Nouhuys/RBMS; Sub-Collections: RADC
RBMS; Clerk: JPC;

RBP 18-MAY-72 11:26 10545

TPO Briefing/5550

Tom, please find out out if I am required to prepare briefing charts on P5550 for Frank's briefing. If so, please supply the format. 1

(J10545) 18-MAY-72 11:26; Title: Author(s): Roger B. Panara/RBP;
Distribution: Thomas J. Barcalow, John L. McNamara/TJB JLM;
Sub-Collections: RADC; Clerk: RBP;

POD MINUTES

The following is output from Redwood POD's meeting of 17-MAY-72. The group seemed to feel that the time was spent productively. 1

Some discussion was devoted to the topic of meetings -- that there are so many and that some seem so unsatisfying to the participants. 2

Some suggestions were proposed that might improve situations: 3

(1) The purpose of a meeting should be clearly stated by its originator, and be understood by all invitees beforehand. 3a

Simply stating the subject of the meeting is usually insufficient. For example, a meeting whose stated topic is the NLS File System might be: 3a1

(a) a tutorial intended for those with no knowledge of the System, 3a1a

(b) a working session intended for those who are responsible for maintaing the File System, or 3a1b

(c) a long-range planning activity for the design of a new File System. 3a1c

Clearly, for an individual to determine whether his attendance is appropriate, he must know which of the three flavors of meeting is actually intended. 3a2

Furthermore, that information is important in his determination of the preparation appropriate. 3a3

(2) The meeting chairman should assume responsibility for assuring that the announced topic and scope are adhered to, since individuals have chosen to come or stay away on the basis of that information. 3b

(3) Any written material that is to be discussed at a meeting should be distributed to the attendees WELL BEFORE the meeting is held. It is ludicrous to hand out documentation at the start of a meeting and then expect intelligent discussion of it after barely enough time to scan its contents. 3c

Each attendee should assume responsibility for reading the appropriate documentation before appearing at the meeting. 3c1

Failing that, the others present should be able to distinguish those people who have read the material from those who haven't, so they can place the proper weight on the comments of each individual. 3c2

/Requiring people to wear badges was suggested as a means for distinguishing the readers from the non-readers. The good guys might wear green badges, and the bad buys red./ 3c2a

Attendees should then view themselves as either active or passive participants, depending upon whether they have come prepared or not. 3c3

(4) The voting machine present at the first meeting of FRAMAC was thought to be a potentially useful tool for helping the attendees to assure that their time is not mis-spent. 3d

(5) Several members of the POD felt that assuring that every other week was free of regularly scheduled meetings (e.g., FRAMAC, PODAC) would help them to increase their productivity, by giving them a full week of uninterrupted work (disregarding the inevitable, spontaneous working meetings that arise, which shouldn't be discouraged). 3e

(6) Finally, one might do well to consider Journal memos as substitutes for meetings, in those cases where the planned activity is sufficiently non-participatory and involves only the dissemination of information from one indiviual to a group. 3f

In line with this thought, and because the number of Journal items that one must regularly peruse is already large, it was suggested that Journal-entry submitters exploit the oft neglected COMMENTS field to classify the entry -- immediate action, information only, meeting announcement, etc. If a standard set of keywords were adopted, then an individual could 'sort' his mail in a way that facilitates its absortion (or discard, as the case may be). 3f1

Some sort of more personalized classification might be possible. For instance, a meeting announced via the Journal to a group of people might be very important to some recipients of the announcement and only of marginal intrest to others. If some formal mechanism existed for distinguishing this type of sub-classification, it would provide another handle for managing one's mail. 3f2

There was some discussion (as there always seems to be) of

PODAC itself, and, in particular, of criteria for determining its effectiveness. 4

It was suggested that it is important that at some point in time there be a judgment made about the effectiveness of POD activity. 4a

Presumably, PODAC exists to (at least ultimately) increase or improve some TANGIBLE output of the group (ARC) by increasing ARC's effectiveness as a team. This improvement may involve: 4a1

the improvement of NLS, 4a1a

its replacement with something superior, 4a1b

its implementation on other types of machines and/or its exportation to other installations, 4a1c

the production of improved hardware systems, 4a1d

cooperation with other system-building groups, 4a1e

the service the NIC provides to the Network community, 4a1f

whatever. 4a1g

If in fact PODAC is intended to improve the effectiveness of ARC in one or more of its formal pursuits, then surely it is important to be able to judge how effectively PODAC as formulated is bringing us to that goal. 4a2

It was suggested that PODAC seek professional assistance in formulating a set of improvement-determining criteria and in applying them (at the approprate time) to assess PODAC's effect on ARC's performance. 4a3

PMK 18-MAY-72 13:17 10548

Change in Network Liaison

Jeanne:
Please change the Network Liaison at MITRE from Peggy Karp to
Jerry Powell. Same address same extension (x2391).
Thanks,
Peggy 1

For the past year UCSB has provided free use of the UCSB On-line System to Network sites. Recharges for this service are being paid for by the Computer Research Lab with funds out of our current ARPA contract. Connect time for the month of April totaled 152 hours. 1

Presently, all users are running on the same account: User Number=196, ID Code=57372, User Name=ARPA. As such, it is difficult to monitor the amount of use by each site. Consequently, we are changing the login procedure for Network users, effective 1 June 72. 2

Commencing June 1, the new login parameters will be: 3

User Number=196 3a
ID Code=57372 3b
User Name=<site designation> 3c
Job Name=<your name> (NIC ID will suffice). 3d

The appropriate site designation is listed at the end of this RFC for each affilliation presently connected to the ARPA Network. Any site omitted from this list which plans on running at UCSB should contact me at (805) 961-3793. 4

Network users of the UCSB On-line System are reminded that these free accounts are provided for Network experimentation. We also encourage use of this service on a casual basis inorder that users unfamiliar with the Culler-Fried System may acquaint themselves with some of the facilities available. However, sites intending to use the system on a production basis are requested to open a private account to which computer charges can be billed. 5

We will also continue to provide free remote batch support on a limited basis. Again, this service is provided as a convenience to Network sites who are attempting to implement and debug Network software. Sites submitting production jobs will be expected to open a Computer Center account and pay for computer charges billed to that account. 6

Affiliation	Network Address	Site Designation (User Name)	7
AMES-67	16	AMES	7a
AMES-ILLIAC	?	AMES	7b
AMES-TIP	144	AMES	7c
BBN-NCC	5	BBN	7d
BBN-TENEX	69	BBN	7e
BBN-TENEXB	133	BBN	7f
BBN-TESTIP	158	BBN	7g
BURR	15	BURR	7h
BURR-TEST	79	BURR	7i
CASE-10	13	CASE	7j
CMU-10	14	CMU	7k
ETAC-TIP	148	ETAC	7l
GWC-TIP	152	GWC	7m
HARV-1	73	HARV	7n
HARV-10	9	HARV	7o
HARV-11	137	HARV	7p
ILL-ANTS	12	ILL	7q
ILL-CAC	76	ILL	7r
LL-67	10	LL	7s
LL-TSP	138	LL	7t
LL-TX-2	74	LL	7u
MCCL-418	22	MCCL	7v
MIT-AI	134	MIT	7w
MIT-DMCG	70	MIT	7x
MIT-MULTICS	6	MIT	7y

MITRE-TIP	145	MITRE	7z
NBS-CCST	19	NBS	7a*
NBS-TIP	147	NBS	7aa
NCAR-7600	25	NCAR	7ab
NCAR-TIP	153	NCAR	7ac
RADC-645	18	RADC	7ad
RADC-TIP	146	RADC	7ae
RAND-CSG	71	RAND	7af
RAND-RCC	7	RAND	7ag
SDC-ADEPT	8	SDC	7ah
SRI-AI	66	SRI	7ai
SRI-ARC	2	SRI	7aj
SU-AI	11	SU	7ak
TINK-418	21	TINK	7al
UCLA-CCN	65	UCLA	7am
UCLA-NMC	1	UCLA	7an
UCSB-75	3	UCSB	7ao
USC-44	23	USC	7ap
USC-TIP	151	USC	7aq
UTAH-10	4	UTAH	7ar

Frameworks and Technical Publications

One major usefulness of technical publications is to reset information--results-- from the spcialized frameworke where they often appear into a more generalized framework. 1

Thus, if we speak of one document replacing another rather than "obsoleteing" we not only avoid grammer that puzzles most readers ("obsolete" is not a verb in standard English) but also open a channel to the whole human experience of one thing replacing another. 2

Similarly the framework of a page indented stepwise on the left and hedged with sometimes lengthy alphanumeric strings on the right, puzzles readers. 3

> The sentence above illustrates a little the results of talking about familiar things in more general frames. 3a

We may on occasion have good reasons to try to force our readers into frameworks that are uncomfortable to them, but in doing so We must recognize that we will pay a price in misunderstanding and conflict. 4

Because of that price, I always start with the assumption that technical publications should be cast in a frame as wide as possible without distoriting the content. 5

6

Network Utility Seminar

MEETING ANNOUNCEMENT

Network Utility Seminar

Several people have suggested that they would find useful, a short seminar on some of the Network-related TENEX utilities. 1

Accordingly, such a tutorial will take place this Friday, 26-MAY-72, between 1 and 2 PM. 2

Discussed will be at least the following: 3

COPYing files between two TENEX systems in the Net 3a

Use of TELNET in accessing other time-sharing systems in the Net 3b

Any other, Net-related topic about which people are curious is also an appropriate topic for discussion and will be dealt with upon request. 4

5

FRAMAC NOTES MAY 19 - RWW 1

Bart Cox chaired the first hour. 2

Bart started off the session, read the two high level goals of the 1962 report. 2a

to find the factors which limit an individual's ability to handle complex problems 2a1

develop systems to better match individual capabilities to problems 2a2

He then asked how people felt about these. The remainder of the meeting was a long rambling discussion, but my interpretation of it was that the following points were at issue with the group generally split. 2b

1. Whether or not very general goals such as above have any real operational meaning. 2b1

2. Why hasn't the group really attacked problem #1 above. 2b2

3. Whether or not instead of such general goals we shouldn't have more concrete applicatons from which to obtain feedback on our progres. 2b3

A vote was taken and the group was split with a small majority for taking more concrete applications dealing with end users, the others more for ARC's historical approach of self-use. 2b3a

There was considerable discussion about evolution in general and ARC's role in this process. Some discussion of ARC being spread too thin to do both good augmentation reserch and applications such as NIC. Some discussion that applicatons such as NIC are really research on another level rather than the classical university type stuff. 2b4

Some concern that our leader didn't really know where we were going or if he did couldn't communicate adequately to lead. 2b5

No general concensus resulted on any point except what felt to me as general agreement that things as now existing were not satisfactory to anyone. 2b6

Those interested in detail who weren't there are referred to the tapes of the session which captured all but the first 15 minutes or so. These tapes may be transcribed. 2b7 more on Output Device Teletype

dick, i am almost positive that i had my paper in right and so forth. i do use a model 37 tty for most of the hardcopy interaction i do, while i use an imlac simulatiing a 37 for most of my editing. there is one thin that would make my use of "o d t" easier -- giving me some time to get the top of page lined up just after the first set of dashes is typed. a question -- why does "odt" send 4 or 5 <DEL> characters at the beginning of each line ? 1

REQUEST FOR OLD RFC'S

NICnote: 1

to: Cindy Page 1a

subject: Old RFC's 1b

date: 11 May 72 1c

from: Buz Owen 1d

Please send me the following RFC's: 2

(RFC#/NIC#) 2a

1/4687 2b

2/4688 2c

3/4689 2d

9/4695 2e

18/4720 2f

25/4727 2g

28/4730 2h

29/4731 2i

33/4735 2j

34/4736 2k

38/4740 2l

39/4741 2m

40/4742 2n

45/4747 2o

46/4748 2p

47/4749 2q

48/4750 2r

49/4728 2s

50/4751	2t
53/4755	2u
54/4756	2v
55/4757	2w
56/4758	2x
57/4759	2y
58/4760	2z
60/4762	2a*
64/4964	2aa
65/4965	2ab
66/5409	2ac
72/5415	2ad
73/5416	2ae
76/5180	2af
83/5621	2ag
86/5631	2ah
94/5725	2ai
95/5731	2aj
98/5744	2ak
100/5761	2al
104/5768	2am
111/5815	2an
117/5826	2ao
122/5834	2ap
125/5841	2aq

126/5842	2ar
130/5848	2as
134/6711	2at
144/6729	2au
146/6742	2av
154/6759	2aw
162/6774	2ax
166/6780	2ay
169/6789	2az
174/6799	2b*
177/7102	2ba
178/7118	2bb
179/7119	2bc
184/7128	2bd
186/7130	2be
187/7131	2bf
190/7135	2bg
191/7136	2bh
192/7137	2bi
194/7139	2bj
195/7140	2bk
199/7151	2bl
204/7196	2bm
206/7176	2bn
208/7181	2bo

209/7187	2bp
210/7189	2bq
215/7545	2br
217/7547	2bs
219/7549	2bt
225/7624	2bu
226/7625	2bv
227/2631	2bw
229/7646	2bx
230/7647	2by
231/7648	2bz
232/7649	2c*
233/7650	2ca
236/7661	2cb
237/7662	2cc
239/7664	2cd
241/7671	2ce
242/7672	2cf
246/7687	2cg
247/7688	2ch
250/7691	2ci
251/7692	2cj
254/7695	2ck
263/7811	2cl
264/7812	2cm

265/7813 2cn

268/7816 2co

269/7817 2cp

273/7837 2cq

278/8056 2cr

280/8060 2cs

281/8163 2ct

282/8164 2cu

283/8165 2cv

285/8271 2cw

286/8272 2cx

289/8295 2cy

290/8300 2cz

292/8302 2d*

294/8304 2da

295/8355 2db

296/8484 2dc

297/8485 2dd

299/8487 2de

301/9073 2df

302/9074 2dg

304/9077 2dh

305/9078 2di

307/9258 2dj

308/9259 2dk

311/9341 2d1

HMM 3ANDAHALF ALPHABETS LOOKS LIKE A LOT AT ONCE PER LINE, BUT IT
WILL BE GREATLY APPRECIATED....BUZ 3

1971 Report to Rome: (documentation, section-I,) reapportioned

As some of you know (documentation, section-I,) has gone bad repeatedly. 1

I have reaportioned the contents into the files named in the links in the following outine: 2

I Team Augmentation 2a

DSS (WSD) <JCN> Pages =10 2a1

(documentation,sec-i-dss,) 2a1a

(documentation,sec-i-dss,) 2a2

Handbook (MFA) <JCN> 2a3

ds 2a3a

BRS (JCN) <DVN>Pages= 5 2a4

(documentation,sec-i-brs,) 2a4a

Basic NLS <JCN> 2a5

(documentation,sec-i-nls,) 2a5a

Internal organization 2a6

(documentation,sec-i-org,) 2a6a

2a7

The file (documentation, section-i,) is still around with links to the contents. 3

When you enter completed drafts, please enter them in the scattered files. 4

DVN 19-MAY-72 16:33 10558

<NIC> Files are Never Private

This message follows DCE's memo on privacy of files (journal,10452).All files in the directory <nic> are open for anyone to read at any time. 1

DVN 19-MAY-72 16:41 10559

My Files Are All Open

This message follows DCE's memo on privacy of files (journal,10452,).Fell free to read my files,any file that appears in my directory is there for your reading or copying. 1

(J10559) 19-MAY-72 16:41; Title: Author(s): Dirk H. van Nouhuys/DVN; Distribution: James E. White, Augmentation Research Handbook, Jacques F. Vallee, Diane S. Kaye, Paul Rech, Michael D. Kudlick, Donald R. Cone, Don Limuti, William R. Ferguson, Priscilla Lister, Linda L. Lane, Marilyn F. Auerbach, Walt Bass, Mary S. Church, William S. Duvall, Douglas C. Engelbart, Beauregard A. Hardeman, Martin E. Hardy, J. D. Hopper, Charles H. Irby, Mil E. Jernigan, Harvey G. Lehtman, Jeanne B. North, James C. Norton, Cindy Page, William H. Paxton, Jeffrey C. Peters, Jake Ratliff, Barbara E. Row, Ed K. Van De Riet, Dirk H. van Nouhuys, Kenneth E. Victor, Donald C. Wallace, Richard W. Watson, Don I. Andrews, Rome Air Development Center (ISIM)/SRI-ARC RADC; Sub-Collections: SRI-ARC RADC; Clerk: DVN;

L10 Changes

Two features of relatively limited utility have been added to L10. 1

(1) UND-OK 1a

If this statement appears in your L10 User Program L10 will not report undefined symbols as errors, thereby allowing you to execute a program which has undefined symbols in it. 1a1

The statement may appar anywhere where a DECLARE is legal, and may be optionally followed by a semi-colon. 1a2

(2) NLSSYM 1b

If this statement appearsf in your program which is being compiled usig output compiler, L10 will define all undefined NLS symbols with the values used in the current system. 1b1

This feature is useful for debugging and loading parts of NLS which are relatively independent of other parts, and can be loaded (at least temporarily) in transient areas. 1b2

DHC 22-MAY-72 17:29 10562

Using ATTACH & AT or AFTER?

Hello, again, again, again;

1. When ATTACHing a detached NLS job (at least when doing it thru the NET), NLS seems to assume that you are a Model 33. It does not seem to remember what type of device you were before detaching.

2. Does the Delete Statement mean 'at' or 'after' the address specified. i.e., will 'd s .1c2' delete .1c2 or .1c3? The Insert command clearly means 'after'. I asked about the possibility oof changing that one when I was visiting for the class. Perhaps I am unique, but I the ambiguity between 'at's and 'after's gets quite confusing for me.

3. Thanx. 1

What to do until CPYNET arrives

LPD 22-MAY-72 21:55 10563

please distribute this further as appropriate

LPD 22-MAY-72 21:55 10563

What to do until CPYNET arrives

This document describes the agonizing procedure for copying files between two TENEXes on the network. 1

Someday there will be a wonderful CPYNET subsystem to do this, or perhaps the file system will be extended to allow opening files at other sites, but for the moment it seems that the existing CPYNET cannot cope with the rapid flux of TENEX versions, geographical barriers, ... 1a

The procedure to be described assumes that you can log in on the host which is to receive the files (called receiver) and then log in on the host which is to send the files (called sender) using TELNET. 2

If you don't know how to use TELNET, type Y when it asks HELP? (at SRI) or type HELP <carriage return> (at BBN). 2a

You must find out your TENEX job number at both ends, using SYSTAT or JOBSTAT. 3

Remember that you can leave TELNET at any time with ↑C and return with CONTINUE if you haven't called another subsystem in between. 3a

Also remember that the job number is printed in decimal; you will have to convert it to octal in a moment. 3b

Let ssn be the number (job # at sender + 303240) * 100000 + 3, all numbers octal, and rsn be the number (job # at receiver + 303240) * 100000 + 2; e.g. if you have job 3 at the sender and job 5 at the receiver, ssn would be 30324300003 and rsn would be 30324500002. 4

Now at the sender give the exec command: 5

`COPY sfile NET:3.receiver-rsn;T` 5a

E.g. 6

`COPY GARBAGE NET:3.SRI-ARC-30324500002;T` 6a

Now immediately do a ↑C and at your own (receiver) end give the exec command: 7

`COPY NET:2.sender-ssn;T rfile` 7a

E.g. 8

`COPY NET:2.BBNTXA-30324300003;T MYGARBAGE` 8a

This will copy the file named sfile at the sender to the file named rfile at the receiver. After a pause the command will complete; do a CONTINUE to get back into TELNET. You should get a @ indicating that the copy completed successfully on the other end. 9

You can give this pair of commands with other files and the same rsn and ssn as long as you do not log out on either end. 10

When you are done, log out in the usual way (i.e. log out at the other end first, then type ↑V↑D to close the connection, then log out at you own end if you wish). 11

TEST

JUST CHECKING TO SEE IF I GET THIS ONLINE OR OFF OR BOTH. BUZ 1

Training: Delays and Plans

Delays: 1

The tutorial workbooks for TNLS did not get in the mail until today. 1a

Duplication of the flip charts will not begin until next week (our only letterer is having finals this week). We should finish them on Friday. 1b

Journal item (journal,8119,) includes an outline of how the charts are arranged. It is a little out of date, but might guide you in assembling them. I intend to put out a new outline as soon as work on the report permits. 1b1

Tom and I should probably have a little computer-augmented dialog about training plans. Unless I hear otherwise, I will put together, again as report work permits, a rough plan of what I think the next training steps should be an we can all look at it. 2

MFA 23-MAY-72 11:36 10566

re: attach/detach and at vs.after

Hi there --
1. Your propensity for embarrassing us has succeeded again. The attach/detach phenomenon (or lack thereof) is a known hassle and will be approached some time or other. To quote the local mavin "it's a can of worms". Usage advice - you takes your chances.
2. This one is easy - Delete statement references the statement AT which the CM is currently positioned. I consider the Insert command to be Insert AT also - even though the way it turns out the new statement is inserted AFTER the address specified. In this case AT an AFTER are pretty much synonymous - nu?
3. yurwelcum 1

meeting annoucment

KEV 23-MAY-72 11:43 10567

There will be a meeting next wednesday (5/31) at 10:30 to discuss the problems of not reading journal mail. The meeting will also be a tutorial as to how to read mail. 1

All those who do not read their mail regularly, (this is principally those who do not work at displays) are strongly encouraged (required?) to attend. 2

If you have any questions regarding this meeting, please address them to JCN or DVN. 3

more on status board

I have not received any comments about my proposal for a system status board. I assume this implies acceptance of the idea and would like any suggestions as to what should be included in the board. 1

Minutes of 16 May 1972 PODCOM Meeting

Attendees 1

Cedar: LLL KEV 1a

Fir: CP WRF (for DCW) 1b

Oak: WLB DL 1c

Redwood: MFA 1d

JDH 1e

Announcements and Requests 2

WRF, MEJ, and WRF requested that they be permitted to participate in the PODCOM meeting in place of the Fir POD representatives. 2a

A motion was made and passed that this not be permitted, and instead WRF was asked to sub for DCW who was out of town. 2a1

There was some feeling that communication was breaking down in Fir POD in that it (they) could not choose representatives who wanted to be on PODCOM and who were willing to try to adequately represent Fir POD there. Some members of PODCOM felt that this was a matter for Fir POD to work out for itself rather than taking up time which PODCOM needed to devote to more general problems. 2a2

Marilyn reminded everyone that Doug has prepared a memo on file-privacy policy (10452,) and that the PODs are expected to discuss this issue and forward their comments and reccommendations to PODCOM. 2b

It was pointed out that Dave's role on PODCOM was still a mystery to many people and that there had been promises that a memo describing his role would soon be forthcoming. Dave explained that it was still a mystery to him as well but that he would try to get out such a memo; Marilyn offered him assistance. 2c

Mike sent a request that the status of the applicants' information blurb be clarified. I (WLB) admitted that I had been sitting on the matter because I didn't have the heart to lay another writing request on DCE, RWW, and JCN who have been valiantly trying to get out a project report and said I thought it would be better for MDK to handle this, as he volunteered to collate the contributions from the several contributors. 2d

Ken reported that Cedar POD has experienced encouraging success with their program of letting a different POD member act as leader each week -- the tone of their meetings has changed from apathy to enthusiasm with just this change in their "modus operandi." 2e

We continued the analysis and planning process which began several meetings ago and reached unanimous agreement on two proposals which, if they are approved by the PODs, will be submitted to Doug as PODCOM recommendations for "official" PODAC policy: 3

PODAC is considered to consist of two complementary thrusts drawing on certain specified resources and coordinated by PODCOM: 3a

The purpose of the first thrust is to provide a representational structure with mechanisms for ARC members to discuss attitudes, issues, and beliefs, and to voice consensus opinions. 3a1

The present structure of PODs, represented on PODCOM will serve as the model for this future structure, with whatever changes are adopted as improvements. 3a1a

The purpose of the second thrust is to make it possible for ARC members to start up and run special interest activities drawing on available PODAC resources. 3a2

PODCOM will act as coordinator for such special interest activities and will consider proposals, allocate resources, and monitor these activities. 3a2a

PODCOM has the authority to allocate up to 10% of ARC's total personnel resources and (within that limit) up to 20% of any individual's time to PODAC activities. 3b

Glossary of 1971 Report to Rome is Ready for Review

A draft of the glossary for the 1971 report to Rome is ready for review (documentation,glossary,). 1

I solicit suggestions about errors or omissions. 2

Marilyn is the reviewer of record, but others may want to glance over the list if only because terms that have been subject to controversey are included. 3

WLB 24-MAY-72 10:14 10572

New Output Processor -- Try It, You'll Like It

REJOICE! REJOICE! The Output Processor has risen again!

The running version of the Output Processor (<subsys>outprc.sav;69>) lacks many of the bugs possessed by its predecessors, notably the notorious randomly-stopping-on-long-files bug, along with several other bugs of more recent vintage. Please let me know if you encounter any bugs in the OP after receiving this message. [If it is necessary to back up to a previous version of the Output Processor, use <subsys>oldoutprc.sav;1.]

Good Luck -- Walt 1

DVN 24-MAY-72 14:17 10573
Documentation Production and Control System Team Meeting of May 23: Minutes

entry 1

(Bass) Rough Outline- Latest Master Catalog Update and
Production System Design (with 3 29 OCT 71 7894 Bass 1a

(Bass)* CATALOG PRODUCTION AUTOMATON DESIGN CHANGE PROPOSAL
4 Aug 71 7465 Bass 1b

(Bass)* CATALOG PRODUCTION AUTOMATON DESIGN PROPOSAL
23 Jul 71 7451 Bass 1c

(Bass)* Catalog Production Automaton Post-Mortem Plans
10 Aug 71 7483 Bass 1d

(Bass)* Catalog Producion Automaton Post-Mortem Plans
10 Aug 71 7483 Bass 1e

(Bass)* Catalog Support System Implementation Plan
15 Dec 71 8005 Bass 1f

(Bass)* Draft of Catalog Production Automaton Preliminary
Design 14 Jul 71 7413 Bass 1g

(Bass)* Master Catalog Entry System Design Proposal
4 Nov 71 7938 Bass 1h

(Bates)* Notes on Hardcopy Equipment
12 Jun 71 7273 Bates 1i

(Duvall)* More on dynamic NLS Documentation
18 Nov 71 8068 Duvall 1j

(Engelbart)* APPENDIX C: Documentation Production and Control
System 13 Jun 71 7281
Engelbart 1k

(Engelbart)* ARC/IPT Project-Continuation Thinkpiece
13 Jun 71 7271 Engelbart 1l

(Engelbart)* Contact Report: DCE visit to Loren Bright, NASA
Ames 25 May 71 7023
Engelbart 1m

(Engelbart)* Note on future sales-type services from NIC and
RINS, and accounting-system 26 Aug 71 7608
Engelbart 1n

(Engelbart)* Notes on Possibility of ARC Giving System Support

to Other Sites' Documentation 21 Jun 71 7306
Engelbart 1o

(Engelbart)* On Engelbart/Balzer discussions regarding his use
of NLS for the Automatic 6 Jan 72 8373
Engelbart 1p

(Engelbart)* Phone Log: DCE to Bill Jones, NASA I4 group.
8 Nov 71 7945 Engelbart 1q

(Engelbart)* Phone log: 6 Oct 71 with Bill Jones, NASA I4
group re their documentation 14 Jan 72 8504
Engelbart 1r

(Engelbart)* Rough Notes on Possibility of ARC Giving System
Support to Ames ILLIAC 18 Jun 71 7294
Engelbart 1s

(Engelbart)* Some Miscellaneous Planning Notes
21 Jun 71 7307 Engelbart 1t

(Watson)* Note On Catalog Production Problem
14 Apr 71 6225 Watson 1u

(Watson)* Requirements and Design Outline for a Stage 1 Master
Catalog Entry and Production 9 Sep 71 7632 Watson 1v

Ron, I got the DRS documentation. Thanks. Jim. 1

10575

<KJOURNAL>10575.NLS;1, 19-AUG-72 2:23 XXX ; Title: Author(s): Augmentation Research Handbook, Kirk E. Kelley, N. Dean Meyer, Kay F. Byrd, Ralph Prather, James E. (Jim) White, Jacques F. Vallee, Diane S. Kaye, Paul Rech, Michael D. Kudlick, Don Limuti, William R. Ferguson, Linda L. Lane, Marilyn F. Auerbach, Walt Bass, Douglas C. Engelbart, Beauregard A. Hardeman, Martin E. Hardy, J. D. Hopper, Charles H. Irby, Mil E. Jernigan, Harvey G. Lehtman, Jeanne B. North, James C. Norton, Cindy Page, William H. Paxton, Jeffrey C. Peters, Jake Ratliff, Barbara E. Rów, Ed K. Van De Riet, Dirk H. van Nouhuys, Kenneth E. Victor, Donald C. Wallace, Richard W. Watson, Don I. Andrews/SRI-ARC; Distribution: Duane L. Stone/DLS(hard copies are bieng air mailed today); Keywords: team journal baseline record system BRS NLS online system PODAC NIC ARC ARPANET offline protocol PDP-10 Imlac Handbook locator Network Resource Notebook plans; Sub-Collections: SRI-ARC; Clerk: DVN;
Origin: <DOCUMENTATION>SPQR.NLS;2, 17-AUG-72 16:57 NDM ;;

ABSTRACT 1

TIME COVERED 1a

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

TENEX 1 B

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

NETWORK INFORMATION CENTER 1c

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

HARDWARE 1d

We have added RP02 disc packs and leased more 30-character-per-second thermal printing terminals and compatible cassette recorders to store text for later processing in the system. 1d1

NEW FEATURES IN NLS 1e

To our online system we have added: 1e1

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

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

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

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

We have begun the redesign of NLS in modular units which will, among other things, ease transfer of all or part of NLS to other systems. 1e2

MANAGEMENT SYSTEMS 1f

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

CONTENTS

Content Statement Number

SUMMARY 3

I. INTRODUCTION 3a

A. 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 DARPA Computer Network. This document reports hardware and software development and applications in several areas, and summarizes plans for continuing development. 3a1

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

C. ARC has begun to maintain online a detailed description of the current state of its activities, the Handbook discussed below, see --,5b). 3a3

1. 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. 3a3a

2. The detailed accounts are in the following Handbook documents: 3a3b

TREE META (10869,), 3a3b1

DEX USER GUIDE (9934,), 3a3b2

DNLS PRELIMINARY USER GUIDE (10703,), 3a3b3

NIC TNLS USER GUIDE (7470,), 3a3b4

NIC JOURNAL USER GUIDE (7635,) 3a3b5

L10 PROGRAMMING GUIDE (USER GUIDE) (9246,) 3a3b6

L10 - A Programming Language for the Augmentation Research Center (Systems Programmer's Guide) (7052,) 3a3b7

D. To take advantage of the automatic reference search of our online system, bibliographic citations in this report are a little unusual looking. They will appear in two forms: 3a4

1. "See-- + a comma + a string of numbers and letters + plus a right parenthesis" [e.g., See--,1a3)] cites some other part of this report as identified by the statement numbers printed right. Online, a reader may cite such an address and move automatically to the appropriate part of the report. 3a4a

2. A four- or five-digit number in parenthesis [e.g. (8277,)] cites a document in ARC's collection. The number is the ARC catalog number. Most of the documents cited in this report are online and an online reader may move to that file automatically as above. A reference section at the end of each chapter supplies bibliographic information about these documents in the usual way. 3a4b

II. HIGHLIGHTS OF 1970 3b

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

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

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

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

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

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

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

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

H. In 1970 we established a collection of documents that form the basis of the Network Information Center, established online techniques for handling the documents, and, most important, began working dialog with the other centers. The combination of our reference data storage techniques with our programming allows retrieving documents according to a variety of attributes and combinations thereof; e.g., year of publication combined with author, or sponsoring institution. We organized with the other sites on the Network to establish Station Agents to handle their interaction with the Network Information Center and supplied the Station Agents with a catalog of their collection and other working materials. To stimulate dialog, pending full operation by connected computers, we set up a central telephone exchange and a system for circulating documents and memos by U.S. Mail through the NIC, including an intra-Net document numbering system. 3b8

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

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

III. HIGHLIGHTS OF 1971 3c

A. Team Augmentation 3c1

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

2. Dialog Support System 3c1b

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

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

Since it became operational in April, 1971, approximately 1200 documents have been generated at ARC and submitted to the Journal. 3c1b1a1

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

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

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

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

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

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

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

These statements contain a link to the document, along with the sender's identification, date/time of submission, document number, and title. 3c1b3b1

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

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

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

3. Handbook 3c1c

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

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

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

4. Baseline Record System 3c1d

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

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

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

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

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

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

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

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

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

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

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

Although we have started using ARC's Baseline Record System on a current task-by-task basis during 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. 3c1d6

5. Basic NLS 3c1e

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

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

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

TNLS and DEX 3c1e2

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

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

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

Changes that make possible cross-file editing have made every word stored on the (1-million word) disks available for manipulation in a given command. 3c1e3

In TNLS addressing by links makes it possible; 3c1e3a

in DNLS split screens whereupon different files may be displayed and their contents exchanged. 3c1e3b

Viewspecs make possible transfer among files of information filtered in various ways. 3c1e3c

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

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

Language development has continued. 3c1e5

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

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

6. Internal Organization 3c1f

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

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

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

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

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

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

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

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

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

PODAC institutionalizes continuing personal and organizational development. 3c1f1c2c

B. Network Information Center: Operations and Development 3c2

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

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

We see the Network Information Center (NIC) as one part of the ARPANET experiment that is interested in the latter problems see--,3c2a1). 3c2a2

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

The NIC is not a classical information center because it provides a wider range than bibliographic and library services. 3c2a3

2. The NIC Public 3c2b

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

Our initial analysis showed us four main needs: 3c2b2

Reference and General Network Information; 3c2b2a

Collaboration Support; 3c2b2b

Document Handling and Creation; and 3c2b2c

Training. 3c2b2d

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

3. NIC Services 3c2c

The initial NIC services now available through the Net to meet the above goals are: 3c2c1

Online: 3c2c1a

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

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

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

Offline: 3c2c1b

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

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

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

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

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

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

(b) ARPA Network Resource Notebook. 3c2c1b2b

(c) Directory of Network Participants. 3c2c1b2c

(d) NIC User Guide. 3c2c1b2d

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

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

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

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

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

4. NIC Goals 3c2d

In the course of its evolution, the ARPANET will continue to generate needs for new software services in interactive data management. 3c2d1

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

C. Network Participation 3c3

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

D. Computer Facility 3c4

1. Hardware 3c4a

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

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

Our hardware configuration contained a number of old, one-of-a-kind pieces of equipment brought over to the PDP-10 system from the previous XDS-940 system. These pieces of equipment have proven difficult to maintain and studies were launched on how to replace or upgrade this equipment. A new BBN network interface and a new DEC R-02 disc system were installed in the spring of 1972, replacing older unreliable equipment. Hardware upgrading of our display system and its special core box has begun to provide temporary relief until a replacement system can be planned. An additional 32k of core is to be added shortly. Studies leading to recommendations to add another channel, disc controller and set of disc drives have been completed. These additions will provide more file space and backup swapping capability. Improved reliability should begin to be manifest in the summer of 1972. 3c4a3

2. System Software 3c4b

TENEX 3c4b1

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

Within the system: 3c4b1b

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

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

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

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

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

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

In the User's View 3c4b1 C

We have set up an advise command so one terminal may control a job loaded at another terminal. 3c4b1c1

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. 3c4b1c2

SUPERWATCH 3c4b2

To help find out what is going on within our timesharing system we have developed an information gathering and formatting program called Superwatch. 3c4b2a

In general Superwatch has been valuable: 3c4b2b

To verify that the system is working as designed. 3c4b2b1

To identify the cause of poor service at the time it is happening (e.g. a bug, hardware malfunction, or just overloading). 3c4b2b2

To identify the "weak link" in the system configuration (drum, disk, memory or CPU capacity). 3c4b2b3

To evaluate changes in the system or hardware configuration. 3c4b2b4

E. Plans for the Future 3c5

1. 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: 3c5a

Advancing the techniques available to ARC and Network system builders and users for augmenting the development and application of computer-based information systems. 3c5a1

Making the Network Information Center into both: 3c5a2

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

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

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

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

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

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

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

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

4. and wherein we become ever better at 3c5d

(4) Operating a utility service. 3c5d1

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

IV. REFERENCES 3d

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

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

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

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

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

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

(10703,) Augmentation Research Center, Stanford Research Institute, Menlo Park, California 94025. DNLS Preliminary Reference Guide. 21 June 1972. Separately paged. 3d7

(10869,) D. I. Andrews, H. G. Lehtman, W. H. Paxton (SRI-ARC). Tree Meta - A Metacompiler for the Augmentation Research Center. In process. Unpaged. 3d8

TEAM AUGMENTATION

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

I. JOURNAL 4a

A. Introduction 4a1

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

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

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

Perhaps he was there at Time T, but 4a1b1a

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

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

Perhaps, although Persons A and B knew he would later need the information, they didn't want to interrupt their own working sequence with the procedure of interrupting Person C and getting him involved. 4a1b1c

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

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

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

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

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

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

Our DSS will provide the following general online aids: multiwindowed displays; simultaneous and independent mobility and view control among many files; link-setup automation; back-link annunciators and jumping; aids for the formation, manipulation, and study of sets of arbitrary passages from among the dialog entries; integration of cross-reference information into hardcopy printouts. 4a1e1

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. 4a1e2

B. The PDP-10 Journal 4a2

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

C. User appearance 4a3

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

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

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

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

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

Clerk: Person actually submitting the document. 4a3b2

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

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

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

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

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

The subcollections listed here are in addition to: 4a3b7a

Any subcollections associated with the submitter by default. 4a3b7a1

Any groups ingluded in the distribution list. 4a3b7a2

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

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

3. Additional to the parameter specification commands are: 4a3c

Commands for control 4a3c1

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

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

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

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

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

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

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

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

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

D. Identification System 4a4

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

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

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

4. 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/j. 4a4d

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

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

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

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

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

E. Number system 4a5

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

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

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

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

F. Document Access 4a6

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

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

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

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

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

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

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

The following is an example of a Journal author index: 4a6b4

* Title	Date	Number	Author	
				4a6b4a
* Scheduler Operation	20 Oct 71	7845	Andrews	4a6b4b
* Proposed new JSYS for TENEX	28 Oct 71	7574	Andrews	4a6b4c
* Proposed Outline for TENEX Documentation	28 Oct 71	7573	Andrews	4a6b4d
* The TENEX Scheduler	21 Jul 71	7419	Andrews	4a6b4e
* response memo	21 Jul 71	7415	Andrews	4a6b4f
* response memo	12 Jul 71	7394	Andrews	4a6b4g
* No title	14 May 71	6982	Andrews	4a6b4h
* known performance problems	28 Feb 72	9313	ARC	

```
                                                 System
                                                 Mea
                 To: ASMG
   ( 9313,1:wznC)                                                  4a6b4i

* SCHED1 problem
18 Feb 72   9232    ARC
                                                 System
                                                 Mea
                 To: ASMG
   ( 9232,1:wznC)                                                  4a6b4j

* re: update
                              5 Jun 72  10633    Auerbach
                 To: DHC
   (10633,1:wznC)                                                  4a6b4k

* let me tell you about the Handbook
                              5 Jun 72  10632    Auerbach
                 To: DHC
   (10632,1:wznC)                                                  4a6b4l

* re: attach/detach and at vs.after
                             23 May 72  10566    Auerbach
                 To: DHC
   (10566,1:wznC)                                                  4a6b4m

* re: output processor directives (again)
                             11 May 72  10417    Auerbach
                 To: DHC
   (10417,1:wznC)                                                  4a6b4n

* Header Positioning Options
                             11 May 72  10413    Auerbach
                 To: DHC
   (10413,1:wznC)                                                  4a6b4o

* re: <documentation>request
                              8 May 72  10384    Auerbach
                 To: DHC
   (10384,1:wznC)                                                  4a6b4p

* PRIMER DRAFT
                              4 May 72  10332    Auerbach
                 To: RWW(please comment) JCN
   (10332,1:wznC)                                                  4a6b4q
```

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

Any Journal document may be located by using the master catalog number as a file name. 4a6b5a

Regardless of the location of the document, the system will find it and return it to the user as requested. 4a6b5a1

At the present time, all recent and most earlier key documents are kept online. 4a6b5a2

An archival system is currrently being implemented. 4a6b5a3

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 ?". 4a6b5a3a

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. 4a6b5a3b

An algorithm based on access activity and priority will be used for determining which documents will be kept permanently in direct access storage. 4a6b5a3c

As with the hardcopy collections, author, number, and titleword indices are provided online as an aid to locating documents. 4a6b5b

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. 4a6b5c

G. Document Distribution 4a7

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

2. A user submitting a document may specify recipients by

simply entering an IDENT for said recipient as one of the parameters specified during submission. 4a7b

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

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

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

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

Current delivery options are hardcopy and online. 4a7d1

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

Other options will be provided as they become necessary. 4a7d2a

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

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

Included in the notification are the document author, number, date, and title; any comments or notes associated with the document; and a link locating the document. 4a7d4a

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

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

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

A provision has been made for automatically starting hardcopy production, but is as yet inoperative because of certain system interface problems. 4a7e2a

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

H. Special Features 4a8

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

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

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

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

I. Problems and Comments 4a9

1. Reliability 4a9a

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

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

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

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

Unless the Journal system immediately recognized this fact, any subsequent Journal entries could potentially cause significant scrambling of related data, resulting in numbers being assigned twice, documents being delivered two or more times (or not at all), or documents disappearing. 4a9a4a

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

Whenever the system is restarted, a special verification and repair program is automatically run. 4a9a5a

This program checks the integrity of Journal files, and (if possible) fixes any errors it finds. If an error is found which cannot be automatically fixed, a message is typed on the operator and logging consoles, and the Journal system is locked. 4a9a5a1

Periodically, a background process runs and checks the validity of various files. Again, if any errors are found, the Journal is locked. 4a9a5b

If any file errors are discovered during the submission process, the Journal is locked and any user currently in the process of submitting a Journal document is notified of a file error, and is returned to the NLS command level. 4a9a5c

2. Operations 4a9b

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

When this occurs, the result is frequently a mess that requires several hours of manual fixup to restore the Journal mechanisms to their proper state. 4a9b1a

This creates an environment which makes reliable operation of the Journal system difficult and subject to the whims of a sometimes unmerciful system. 4a9b1b

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. 4a9b1c

The major current cause is running out of Disc space, which TENEX does not handle very gracefully. 4a9b1c1

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

Again, this is in the area of file handling. 4a9b2a

This asynchrony has made consistent production of hardcopy difficult. In fact, for a while it was virtually impossble. 4a9b2b

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. 4a9b2c

J. Summary 4a10

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

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

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

Future efforts will attempt to improve the efficiency. 4a10b2

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

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

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

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

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

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

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

II. HANDBOOK 4b

A. Description 4b1

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

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

Planning -- plans, contingency alternatives, resource commitments, status, criticisms 4b1b1

Designing -- designs, design principles, constraints, estimates, analyses, supportive data, relevant needs and possibilities 4b1b2

Operating -- roles, task definitions, assignments, policies, operational procedures and conventions 4b1b3

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

4. At present, we have just finished the first and very primitive pass at organizing and obtaining in hardcopy much documentation relevant to the contents of an ARC Handbook. An outline is included in this report, see--,12). 4b1d

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

revision/maintenance for the Handbook in its current form. 4b1e

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

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

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

III. BASELINE RECORD SYSTEM 4c

A. Introduction 4c1

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

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

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

ARC planning and task activity is currently conducted in the LINAC operational framework outlined below, see--,5e3). 4c1a3

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

The information relative to the planned system developments is contained in our Baseline Record. 4c1a4a

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

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

1. To provide a central place for recording Baseline data in an organized way. 4c1b1a

2. To prepare useful views of such data. 4c1b1b

3. To provide a system for updating the Baseline data base. 4c1b1c

The main responsibility for the data actually being complete and current resides with the pushers for the various tasks and activities. 4c1b1c1

Some BRS design criteria are: 4c1b2

Users' opinions should be gathered and brought into the BRS system design process as it progresses. 4c1b2a

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. 4c1b2b

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. 4c1b2c

Views must be "easy" to generate - both by the Operations people and by individual ARC users wanting special views. 4c1b2d

Routinely produced views must be meaningful and useful to a wide range of users' needs. 4c1b2e

Users must be guided - trained - in the use of the BRS, probably on a continuing basis. 4c1b2f

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

This record is produced from central planning data contained in online files at ARC, and will contain various views of that information as needed to give meaningful representations of our situation. 4c1b3a

A basic set of Baseline record views we will use includes: 4c1b3a1

(1) Schedule: by activity grouping (NIC,DSS,CSO) 4c1b3a2

(2) Schedule: all tasks by ARC planning stage 4c1b3a3

(3) Schedule: all tasks by person 4c1b3a4

(4) Baseline record summaries by task, formatted as "status" reports, with elements such as: 4c1b3a5

Information: (about nature of task and agreements) 4c1b3a5a

Buyer(s): (for whom or what task is this task being performed) 4c1b3a5b

Requirements: (agreed upon needs this task will fulfill and certain design criteria as needed) 4c1b3a5c

Design: (details of design--or links to such--user interface features, internal implementation) 4c1b3a5d

Milestones: (significant delivery/evaluation points used when relevant) 4c1b3a5e

Subtasks: (smaller segments made visible for more detailed planning purposes as needed) 4c1b3a5f

Subcontracts: (other tasks initiated in direct support) 4c1b3a5g

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

Sections of the Baseline Record that are superseded by new Journal entries will be separately shelved with other obsolete documents. 4c1b4a

Changes in requirements and designs will be approved and recorded as in configuration management of hardware designs. 4c1b4b

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

B. Present Baseline Record System 4c2

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

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

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. 4c2a2

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. 4c2a3

C. Higher-Level Planning Needs 4c3

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

2. Plan needs 4c3b

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

Because our activities are strongly affected by the developments (features, timing, resource use) of others, it is clear that realistic plans for each activity will be produced only after considerable integration and adjustment. 4c3b1a

3. Plans needed and who makes them 4c3c

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

He is expected to draw upon others (including his DSS planning team) for help, ideas, or other inputs in the process, but he is the one responsible for producing the plans we need. 4c3c1a

Rather than just getting help from others individually, he may find it may useful to have some group discussions among appropriate people for each main activity. The pusher should make this happen where needed. 4c3c1b

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

4. Plan elements 4c3d

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

1. Basic objectives of the activity. 4c3d1a

What should it result in or produce? 4c3d1a1

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. 4c3d1b

These may be thought of either as separate tasks, or simply as "features" -- which might result from several tasks. 4c3d1b1

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. 4c3d1c

4. Stages of development -- logical combinations of features, procedures, training (not just points in time, describing the "look" at significant points. 4c3d1d

The stages should fit the natural progression of the activity -- not necessarily related to ARC overall stages. 4c3d1d1

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. 4c3d1d2

5. Relationships to other tasks or features needed. 4c3d1e

Where critical needs (for each activity) exist, they will be pointed out -- with some discussion of the situation. 4c3d1e1

6. Effort needed to meet stages. 4c3d1f

ROUGH estimates in man-weeks by feature or stage (plus skill types or people being considered to work on it if known) are needed. 4c3d1f1

7. Alternative possibilities for other features or stages. 4c3d1g

8. Implications on the staffing skills and levels required of ARC as a whole. 4c3d1h

D. Comments on our Experience with the BRS to Date 4c4

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

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

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

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

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

Developing a system for the facilitation of input of data is a real challenge, but must be worked out. 4c4b3a

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. 4c4b3b

IV. BASIC NLS 4d

A. Basic NLS User Features 4d1

1. Introduction 4d1a

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

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

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

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

the addition of the split screen display mode to make possible multi-file viewing and cross-file editing. 4d1a1a2a

the removal of the trails feature because it was not used extensively. 4d1a1a2b

the modification of the substitute command to provide a larger, more useful variety of parameter modes. 4d1a1a2c

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

Our experiences in the development of augmentation system features within the Center and on the ARPA Network indicate some new directions for our bootstrapped research effort. 4d1a2

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

Such tools include: 4d1a2a1

the Dialog Support System (DSS), and 4d1a2a1a

the Baseline Record System (BRS). 4d1a2a1b

The first scattered community will be composed of system designers aided primarily by the Software Engineering Augmentation System (SEAS) discussed below see--,5d2). This community will collaborate in the development of a system design discipline. The augmentation of the Software Engineering community will accelerate evolution of new tools. In the future, other communities will receive specialized tools developed by the augmented system designers. 4d1a2b

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

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

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

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

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

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

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

Several training courses for Network users of the NIC and TNLS have been held. They are described in this report as part of NIC activities see--,6g10a)). 4d1a3e

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

2. NLS -- Technical overview 4d1b

Introduction 4d1b1

The current implementation of NLS on the PDP-10 is a large, continually evolving program. Code presently occupies about 150,000 words of computer storage. 4d1b1a

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

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

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

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. 4d1b1b2a

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--,5d2f). 4d1b1b2b

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

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

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

DEX: See DEX user guide (9934,) and below see --,5d1d). 4d1b1c3

Journal: NIC Journal User Guide (7635,) and see--,5a) and --,6f2a) of this report. 4d1b1c4

Identification: See TNLS user guide (7470,) and see --,5a4) in this report. 4d1b1c5

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

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

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

NLS-DDT: See --,4b2b1). 4d1b1c9

NLS file structure 4d1b2

Introduction 4d1b2a

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

It is desirable to have virtually no limit on the size of a file. This means it is not practical to have an entire file in core when viewing or editing it. 4d1b2a1a

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. 4d1b2a1b

In executing a single editing function there may be a large number of structural operations. 4d1b2a1c

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

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

File Header 4d1b2b

File header contents: 4d1b2b1

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

Structure Blocks -- ring elements 4d1b2c

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

Ring element contents: 4d1b2c2

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

Data Block -- statement data blocks 4d1b2d

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

Statement Data Block (SDB) header contents: 4d1b2d2

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

String Identifiers and Text Pointers 4d1b2e

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

If the string is in an NLS statement, the STID contains a file identifier and a ring element identifier. 4d1b2e1a

The presence of a file identifier within the STID all editing functions to be carried out between files. 4d1b2e1b

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

Locking mechanism -- Partial copies 4d1b2f

The NLS file system under TENEX provides a locking mechanism, which protects against inadvertant overwrite when several people are working on the same file. Once a user starts modifying a file, it is "locked" by him against changes by other users until he deems his changes consistent and complete and issues one of the commands: Update File, Output File, or Unlock File which "unlock" the file. Note, a user can leave a file locked indefinitely -- this protection is not limited to one console session. 4d1b2f1

When a file is locked (is being modified), the user who has modification rights sees all of the changes that he is making. However, others who read the file will see it in its original, unaltered state. If they try to modify it, they will be told that it is locked by a particular user. Thus the users can negotiate for modification rights to the file. 4d1b2f1a

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

All modifications to a file are contained in a partial copy file. These include modified ring elements and SDB's. 4d1b2f2a

Core Management of File Space 4d1b2g

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

1. Core-resident pages are checked for sufficient free space. 4d1b2g1a

2. Other pages are checked for free space. If one has sufficient space, it is brought in. 4d1b2g1b

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. 4d1b2g1c

4. Otherwise a new page is created. 4d1b2g1d

Logical structure 4d1b3

Introduction 4d1b3a

Figures FIGUREa and FIGUREb represents the logical structure of the NLS system. Major components of the system are discussed below. 4d1b3a1

Interaction support 4d1b3b

Terminal interaction support 4d1b3b1

Display interaction support 4d1b3b1a

The display interaction support routines take input from display users, support various L10 display input constructions which allow the creation of simple interaction statements, and control the command feedback line, name area, view spec area, and bug selection areas of the display screen. 4d1b3b1a1

Typewriter interaction support 4d1b3b1b

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. 4d1b3b1b1

Sequential file input support 4d1b3b2

Sequential file input support routines take input from DEX sequential files or a control file and pass it to the DEX subsystem processor or the control file driver system, respectively. 4d1b3b2a

Subsystem control 4d1b3c

Command specification 4d1b3c1

The command specification routines receive information from the input interaction level or sequential file input and process it as follows: 4d1b3c1a

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. 4d1b3c1a1

2. Operands for commands are interpreted where necessary. 4d1b3c1a2

3. Control is transferred to the appropriate execution routine. 4d1b3c1a3

4. Control is transferred to the Portrayal Generator for formatting and display. 4d1b3c1a4

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. 4d1b3c1a5

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. 4d1b3c1b

Subsystem Support 4d1b3c2

These routines support the parsing of particular subsystems and provide the code necessary to translate the high level functions of each subsystem into calls on the file manipulation and portrayal generation routines of NLS. They also have code necessary to implement any additional facilities needed by the subsystem. 4d1b3c2a

Portrayal generator 4d1b3d

Display control 4d1b3d1

The display controller is composed of 4d1b3d1a

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 4d1b3d1a 1

2) user controls, such as the DNLS jump commands, over what is portrayed and how much is shown. 4d1b3d1a2

This formatter can maintain images in several "display areas" at one time, updating them as necessary. Each area may display information from several files. 4d1b3d1b

Typewriter terminal print control 4d1b3d2

This is a formatter that is oriented toward printing parts of a file onto a typewriter terminal. 4d1b3d2a

Hardcopy formatters 4d1b3d3

These include a relatively simple system, Quickprint, and a more complicated formatting program, the Output Processor. 4d1b3d3a

Quickprint formats the text for printing as it appears through the display or typewriter terminal formatters. 4d1b3d3a1

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", NEW REFERENCE(Journal,11076,2). 4d1b3d3a2

Sequence generator 4d1b3d4

Succeeding calls on the sequence generator create a sequence of statements which satisfy system or user filters starting at a place in the file specified by the user. 4d1b3d4a

An example of the system filters it

observes in deciding whether the identifier of a statement should be part of a sequence is the level truncation viewspec which permits the display of only those statements above particular levels in the NLS hierarchical file structure. 4d1b3d4a1

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. 4d1b3d4b

See--,4d1e4) for a discussion of the sequence generator with user programs. 4d1b3d4c

User filters and reformatters 4d1b3d5

The user may write and incorporate additional filters which the sequence generator will use as a final acceptance test. These user-supplied filters may reformat the text of the file for special applications or views. 4d1b3d5a

User sequence generators 4d1b3d6

The user can write his own sequence generators which can make use of any NLS routines. 4d1b3d6a

Editing 4d1b3e

File manipulation algorithms 4d1b3e1

These algorithms carry out the file manipulation commands of NLS. They decide what is to be done by the textual and structural editing routines and in what order. Utility routines actually manipulate the NLS files. 4d1b3e1a

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"). 4d1b3e1a1

These algorithms can move and copy text from one file to another through cross-file editing.. 4d1b3e1b

Structure editing 4d1b3e2

These routines involve the manipulation of ring structure alone and do not alter the contents of the statement data blocks which contain the text. 4d1b3e2a

Text editing 4d1b3e3

These routines edit the text of NLS statements. Content analysis features of L10 are used to determine where changes should take place; the string manipulation and SDB manipulation machinery then change the contents of the file. 4d1b3e3a

Special purpose processors 4d1b3f

Inserting and outputting sequential files 4d1b3f1

These processors create NLS files from sequential files and vice versa. 4d1b3f1a

Compilers 4d1b3f2

Currently four compilers are available from NLS. In addition we are now studying ways of making available through NLS the assemblers of the TENEX operating system. 4d1b3f2a

The four compilers now available are: 4d1b3f2a1

L10, a procedure-oriented, block structured language developed by ARC for use on the PDP-10, 4d1b3f2a1a

A subset is available as the content analyzer. (9246,10) and see --,4d1e4ca1) in this report.) 4d1b3f2a1a1

IMOL, a procedure-oriented, block structured language which produces code for the IMLAC computer-display. 4d1b3f2a1b

Tree-Meta, a compiler-compiler used by ARC staff to develop other languages, such as L10 and IMOL. (See the Tree-Meta Report (10869,,) and --,4d2e) of this report.) 4d1b3f2a1c

MPL, the Modular Programming Language, an experimental new language to be used to rewrite NLS. (See --,4d2f) 4d1b3f2a1d

Text is passed to these compilers through the sequence generator (and thus can be filtered and reformatted enroute to these various processors). 4d1b3f2b

Utility routines 4d1b3g

NLS file system 4d1b3g1

These routines implement and manipulate the data structures in Tenex files which NLS uses. Unlike other routines discussed above, they are cognizant of and deal with the data structures and the TENEX timesharing system environment. 4d1b3g1a

They are responsible for: 4d1b3g1b

Opening and closing files. 4d1b 3g1b1

Managing the portion of core set aside for file pages. 4d1b3g1b2

Writing on and reading from files. 4d1b3g1b3

Manipulating ring elements and SDB's. 4d1b3g1b4

Moving within the NLS file structure by following ring element pointers. 4d1b3g1b5

Statement name lookup. 4d1b3g1b6

NLS string system 4d1b3g2

Supports string manipulation construcions in the L10 language and deals with the NLS

Statement Data Block and Ring Block structure. 4d1b3g2a

Miscellaneous support routines 4d1b3g3

Basic L10 language support routines. 4d1b3g3a

Call mechanisms. 4d1b3g3a1

Display support routines 4d1b3g4

Information writing on the screen. 4d1b3g4a

Manipulating information on the screen. 4d1b3g4b

Basic input routines 4d1b3g5

Basic typewriter terminal output routines 4d1b3g6

3. NLS -- New features 4d1c

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

Name Delimiters 4d1c2

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

Jump to Word/Content 4d1c3

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

Null File 4d1c4

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

Output Assembler 4d1c5

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

Output Compiler 4d1c6

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

Output Sequential 4d1c7

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

Insert Sequential 4d1c8

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

Output Quickprint 4d1c9

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

Update File -- File Locking 4d1c10

The NLS file system under TENEX provides a locking mechanism, which protects against inadvertant overwrite when several people are working on the same file. Once a user starts modifying a file, it is "locked" by him against changes by other users until he deems his changes consistent and complete and issues one of the commands: Update File, Output File, or Unlock File which "unlock" the file. Note, a user can leave a file locked indefinitely

-- this protection is not limited to one console session. 4d1c10a

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

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

Goto Exec 4d1c11

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

Execute Logout 4d1c12

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

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

Split Screen and Cross File Editing 4d1c14

Display Screen Splitting and Formatting 4d1c14a

Goto Display Area Control 4d1c14a1

Horizontal Split 4d1c14a1a

This splits the display area in which the BUG occurred horizontally (into an upper and lower segment) at the bugged location moving the image of the original display area to the upper or lower segment depending on whether the cursor is above or below the bugged position when the final CA is input. 4d1c14a1a1

No display area will be created which is smaller than 2 lines by 20 columns (using the character size of the original display area). 4d1c14a1a1a

Vertical Split 4d1c14a1b

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. 4d1c14a1b1

Move Boundary 4d1c14a1c

The selected boundary (first BUG) is moved to the new position (second BUG). A boundary will not be moved past a boundary of a neighbor. A boundary is moved for all display areas for which it is a boundary. Any resulting display area which is smaller than 2 lines by 20 columns will be deleted. 4d1c14a1c1

Format Display Area 4d1c14a1d

Character Size 4d1c14a1d1

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. 4d1c14a1d1a

Clear Display Area 4d1c14a1e

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. 4d1c14a1e1

Cross File Editing in DNLS 4d1c14b

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

For example, if one executes a Jump command, the position of the cursor when the final command accept is entered determines in which display area the new image is to appear. 4d1c14b1a

Also, if one changes viewspecs using the leftmost two buttons of the mouse, the viewspecs of the display area containing the cursor when the buttons go down are used as the initial values and are displayed in the viewspec area. When the buttons are released, the display area containing the cursor receives the new viewspecs. 4d1c14b1b

Substitute Command Change 4d1c15

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

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

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

Display Creation Efficiency Improvement 4d1c16

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

4. The Deferred Execution System (DEX) 4d1d

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

The currently running system, DEX-1, has commands that provide for text input, backspacing over characters, deletion (and undeletion) of commands, and the creation of NLS files and hardcopy printouts. DEX-2 will provide further editing capabilities as well as access to existing files. 4d1d2

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

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

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

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

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

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

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

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

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

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

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

5. Other subsystems 4d1e

Sorter-Merger-Updater Description 4d1e1

General Implementation Description 4d1e1a

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

Each of the three primitives added to NLS to perform the sorting, merging, and updating functions requires as an argument the address of a key procedure program written in the L10 language to furnish sorting criteria. In addition, the update primitive requires as an argument the address of an update decision procedure to take action on corresponding data items. This procedure will differ for various specific applications. In the most general case, it is provided by the user, although we are building up a library of some standard procedures for common applications. 4d1e1a1a

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. 4d1e1a1b

Procedures supplied to the Sort, Merge, and Update primitives: 4d1e1b

Key procedure: 4d1e1b1

Sort key procedures are written in the L10 language and provide the patterns for text string analysis through which a data base in an NLS file is to be sorted. 4d1e1b1a

Typical keys may be written to: 4d1e1b1b

find and order last names after initials 4d1e1b1b1

find numbers in columns 4d1e1b1b2

find individual key words in indices 4d1e1b1b3

The system default alphabetizes statements over which the system is run. 4d1e1b1c

Update Decision Procedure 4d1e1b2

The update decision procedure is called by the update primitive once for each sort key value found in either the master or update input. 4d1e1b2a

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. 4d1e1b2a1

In general, this procedure changes the master file by deleting some branches from the master input and inserting some of the update input. 4d1e1b2b

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. 4d1e1b2b1

A comparison file may be created by this procedure for proof reading. 4d1e1b2c

Control File and Record Mode 4d1e2

A set of commands (and modifications to the user input routines) has been added to implement a record and playback capability. A session or series of operations at a display console may be recorded on a file, then played back. During the playback, NLS will read the input from the control file instead of from the user. An attempt is made to replay the commands at the same speed that the user entered them. 4d1e2a

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

Output Processor Addition 4d1e3

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

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

FR-80 Output Processor Device 4d1e3b

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

64 character sizes, 4d1e3b1a

placement of text within a 16k by 16k coordinate system, 4d1e3b1b

various intensities and line widths, and 4d1e3b1c

microfilm/fiche and/or paper output. 4d1e3b1d

User Programs 4d1e4

Introduction 4d1e4a

User-written programs enable one to tailor the presentation of the information in a file to his particular needs. Experienced users may write and compile online programs that edit files automatically. These programs, written in the

L10 programming language used by NLS system programmers, may be composed using the NLS text editor, compiled into the user program buffer, and linked into the user's running NLS system. 4d1e4a1

The language contains some high level features for operations such as string analysis and manipulation which are implemented in the language as calls on NLS library routines. 4d1e4a1a

The User Program facility brings together the tools formerly described as Higher Level Processes (HLPs) in the June 1971 Report (8277,). The current system provides the user with access to the full array of NLS system tools as well as the debugging facility, DDT. The ability to create what are known as User Sequence Generator programs allows greater file reordering than did the old Analyzer Formatter. User Programs also satisfy some objections to the earlier Executable Text, which could not be easily programmed or debugged. 4d1e4a1b

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

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

For example, one may want to see only those statements containing a particular word or phrase. 4d1e4a3a

Or one might want to see one line of text that compacts the information found in several longer statements. 4d1e4a3b

One might also wish to perform a series of routine editing operations without specifying each of the NLS commands over and over again. 4d1e4a4

The Network Information Center at ARC uses the ability to create text using the information from several different statements (and even different files) and the ability to insert this new text into a file to produce catalogs and indices. 4d1e4a4a

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. 4d1e4a5

Users taking advantage of this expanded feature also have access to the debugging facility of the system. Currently this means that the TENEX DDT may be used with compiled and instituted user programs (i.e., those which have been linked into the user's running NLS system). A planned expansion will make available a debugger in the NLS system itself providing an extremely powerful programming tool. 4d1e4a6

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. 4d1e4a7

Some non-programming personnel at ARC, however, have been creating programs to produce formatted catalogs; programmers have used the feature to create and debug new NLS commands and subsystems without being forced to compile and load the entire NLS system whenever a change is made, an inefficient and time consuming process given the demands on system resources and the current size of the system. 4d1e4a8

To make this powerful tool more generally usable, an initial documentation of a subset of the L10 language has been created. This "L10 Primer" provides basic information on the syntax and semantics of many of the constructions of the whole language. It also describes the basic commands in NLS that provide the user interface between NLS and user programs. (9246,). 4d1e4a9

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. 4d1e4a9a

Creation of User Written Programs 4d1e4b

User written programs must be coded in L10. They may call other user written routines and various procedures in the NLS program itself. 4d1e4b1

User programs that control the way material is portrayed take effect when NLS presents a sequence of statements in response to a command like Print Group (in TNLS) or Jump to Item (in DNLS). 4d1e4b2

In processing a command such as Print, NLS looks at a sequence of statements, examining each statement to see if it falls within the range specified in the Print command and if it satisfies the viewspecs. At this point NLS may also pass the statement to a user written program to see if it satisfies the requirements specified in that program. If the user program returns a value of true, the (passed) statement is printed and the next statement in the sequence is tested; if false, the next statement in the sequence is tested. 4d1e4b2a

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. 4d1e4b3

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. 4d1e4b3a

Context Of User Written Programs -- The Portrayal Generator 4d1e4c

Generally, the user written program runs in the framework of the portrayal generator. It may be invoked in several ways, described below, whenever one asks to view a portion of the file, e.g., with a Print command in TNLS, with any of the Output to Printer commands, and with the Jump command in DNLS. 4d1e4c1

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. 4d1e4c2

Sequence Generator 4d1e4c3

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. 4d1e4c3a

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. 4d1e4c3a1

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. 4d1e4c3b

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. 4d1e4c3c

Formatter 4d1e4c4

The formatter arranges text passed to it by the sequence generator (described below) in the style specified by the user. The formatter observes viewspecs such as line truncation, length and indenting; it also formats the text in accord with the requirements of the output device. 4d1e4c4a

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). 4d1e4c4b

User Filters 4d1e4c5

The user filter program may be either a content analysis pattern or a more complex L10 program. 4d1e4c5a

Content Analysis Patterns 4d1e4c5a1

Content analysis patterns describe characteristics that a statement must have to be included in the sequence being generated. For example, a content analysis pattern may stipulate that a statement must contain a particular phrase, or that it must have been written since a particular date. In general, content analysis patterns may use any of the pattern matching facilities permitted in L10 FIND statements. 4d1e4c5a1a

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. 4d1e4c5a1b

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 L10 program. 4d1e4c5a1c

User Written L10 Programs 4d1e4c5a2

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). 4d1e4c5a2a

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. 4d1e4c5a2b

User-Written Sequence Generators 4d1e4c6

A user may provide his own sequence generator to be used in lieu of the regular NLS sequence generator. Such a program may call the normal NLS sequence generator, as well as content analysis filters and user-written L10 programs. It may even call other user-written sequence generators. 4d1e4c6a

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. 4d1e4c6b

Examples Of Content Analysis Patterns And L10 User Programs 4d1e4d

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

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

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

Examples of Simple Content Analysis Patterns 4d1e4d4

BEFORE (25-JAN-72 12:00); 4d1e4d4a

This pattern will match those statements created or modified (whichever happened most recently) before noon on 25 January 1972. 4d1e4d4a1

`ID = HGL OR ID = MFA;` 4d1e4d4b

This pattern will match all statements created or modified (whichever happened most recently) by users with the identifiers "HGL" or "MFA". 4d1e4d4b1

`D 2$LD / ["CA" / "Content Analyzer"];` 4d1e4d4c

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. 4d1e4d4c1

Note the use of the brackets to permit an unanchored search -- a search for a pattern anywhere in the statement. Note also the use of the slash for alternations. 4d1e4d4c1a

`[(2L (SP/TRUE) /2D) D '- 4D];` 4d1e4d4d

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. 4d1e4d4d1

Examples include YU 4-1234, YU4-1234, and 984-1234. 4d1e4d4d1a

Examples of Analyzer-Formatter Programs 4d1e4d5

The following are examples of user analyzer-formatter programs which selectively edit statements in an NLS file on the basis of text searched for by the pattern matching capabilities. Examples of more sophisticated user programs such as sort keys and user sequence generator programs will be presented in a later supplement with a description of NLS routines easily accessed by users. 4d1e4d5a

Example 1-- 4d1e4d5b

```
PROGRAM outname % removes statement names
-- del= () --%                                4d1e4d5b1
   DECLARE TEXT POINTER sf, paf, pae;         4d1e4d5b1a
   (outname)PROCEDURE;                        4d1e4d5b1b
      IF FIND ↑sf $NP '( ↑paf [')] ↑pae
      THEN                                   4d1e4d5b1b1
         BEGIN                              4d1e4d5b1b1a
         ST sf ←  pae SE(sf);               4d1e4d5b1b1b
         RETURN(TRUE);                      4d1e4d5b1b1c
         END                                4d1e4d5b1b1d
      ELSE RETURN(FALSE);                    4d1e4d5b1b2
      END.                                   4d1e4d5b1b3
   FINISH                                     4d1e4d5b1c
```

This program removes any parenthesized expression whose opening parenthesis corresponds to the first printed character of an NLS statement. 4d1e4d5b2

Example 2-- 4d1e4d5c

```
PROGRAM changed                                 4d1e4d5c1
(changed)PROCEDURE;                             4d1e4d5c2
   LOCAL TEXT POINTER f, e;                     4d1e4d5c2a
   FIND ↑f SE(f) ↑e;                            4d1e4d5c2b
   IF FIND SINCE (25-JAN-72 12:00) THEN         4d1e4d5c2c
      BEGIN                                     4d1e4d5c2c1
      ST f ← "[CHANGED]", f e;                  4d1e4d5c2c2
      RETURN(TRUE);                             4d1e4d5c2c3
      END                                       4d1e4d5c2c4
   ELSE RETURN(FALSE);                          4d1e4d5c2d
   END.                                         4d1e4d5c2e
FINISH                                          4d1e4d5c3
```

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. 4d1e4d5c4

B. Software Engineering Augmentation Systems (SEAS) 4d2

1. Introduction 4d2a

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

In this past contract period, we have taken several steps to further augment the software engineer -- in fact, we have coined the acronym SEAS (for Software Engineer Augmentation System) to give specific system orientation towards the end of developing a full and balanced set of tools, techniques, methods, principles, etc. for augmenting software engineers. The developments described below are part of an accelerating activity -- an important part of our near-future plans in the next contract period involve a greater level of activity here. 4d2a2

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

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

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

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

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

2. Source Level Debugging 4d2b

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

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

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

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

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

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

3. System Measurement 4d2c

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

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

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

Measuring the elapsed time between two instructions. 4d2c3

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

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

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

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

Monitor measurements 4d2c5

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

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

4. Source Code Documentation Standards 4d2d

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

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

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

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

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

5. Languages 4d2e

Introduction 4d2e1

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

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

In collaboration with several people at the Xerox Palo Alto Research Center, work has begun on a Modular Programming System (MPS) and a Modular

Programming Language (MPL) that will replace the current languages and in which the NLS system will be redesigned and rewritten for greater efficiency and flexibility. 4d2e1b

L10 4d2e2

NLS on the PDP-10 is written in the L10 programming language, an ALGOL-like language that has some high level special purpose features for string analysis and manipulation and for interacting with NLS users. 4d2e2a

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

Tree-Meta 4d2e3

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

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

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

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

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

6. Modular Programming 4d2f

Goals 4d2f1

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

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

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

(a3) they evolve and grow throughout their lifetimes (and there is evidence that they also "age" [Lehman & Belady]). 4d2f1a3

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

Desirable Characteristics 4d2f2

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

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

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

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

(a4a) a1-a4 imply that software components, 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. 4d2f2c

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

a4a, a5, and a6 then imply that 4d2f2e

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

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

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

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

(a3c) One way of constructing useful components is to build them from combinations of already existing modules (a3b). Hence there must be a way of bundling useful configurations together as seemingly atomic modules so they can be readily reused. 4d2f2h

MPS Capabilities 4d2f3

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

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

Simple function call and return mechanisms alone do not satisfy this requirement. 4d2f3a1a

Data definition facilities that: 4d2f3a2

clarify the specification of the data structures which, together with control, completely specify the interfaces between modules; 4d2f3a2a

are potentially economical in space and accessing speed without being dependent on a particular machine; 4d2f3a2b

are an aid in developing and describing program components and the structure of algorithms. 4d2f3a2c

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

For example, for binding a procedure call to a real procedure, a "typed" pointer to a data structure of the correct type, etc. The set of bindings for a module's virtual objects at a given moment comprises the environment for that module. 4d2f3a3a

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

This enables debugging and measuring tools to be built as standard MPS programs and along with dynamic binding allows such tools to be brought to bear on MPS programs whenever necessary. 4d2f3a4a

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

This allows systems to be partly initialized by partially executing them and then bundling them up for later use with the initialization computations factored out; 4d2f3a5a

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; 4d2f3a5b

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. 4d2f3a5c

V. INTERNAL ORGANIZATION 4e

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

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

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

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

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

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

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

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

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

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

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

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

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

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

B. FRAMAC 4e2

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

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

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

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

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

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

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

Our Second Stage will include: 4e2b3

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

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

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

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

4. About the initial composition of our FRAMAC group: 4e2d

We had been visualizing a small FRAMAC group, considering the type of dialog we hope for. But when we reviewed our LINAC planning-team composition, we decided that there is a such strong interaction between our current planning exercise and our Framework that we couldn't seem to find a logical way to cut the group membership smaller. The initial FRAMAC group numbers nineteen ARC people plus two other SRI management people. 4e2d1

C. LINAC 4e3

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

LINAC serves several basic needs: 4e3a1

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

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

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

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

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

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

Some of our activities will be funded by what amounts to a taxation upon all or some of the projects. Such taxation measures will be established and monitored with due representation by the concerned parties. 4e3a2c

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

OPERATIONS 4e3b1

Administration 4e3b1a
Computer Service Operations - Hardware 4e3b1b
Computer Service Operations - Software 4e3b1c
Computer Service Operations - Operators 4e3b1d
People Service Operations 4e3b1e
User Interface 4e3b1f

DEVELOPMENT THRUSTS 4e3b2

Development Coordination 4e3b2a
Delivery and Marketing 4e3b2b
Dialog Support System (DSS) 4e3b2c
Documentation Production and Support System (DPCS) 4e3b2d
Baseline Record System (BRS) 4e3b2e
System Developers Handbook System (SDHS) 4e3b2f
Software Engineering Augmentation System (SEAS) 4e3b2g
General Development (not included in above thrusts) 4e3b2h

PROJECTS 4e3b3

ARPA/RADC Project: Team Augmentation Portion 4e3b3a
Administration 4e3b3a1
ARPA/RADC Project: Network Information Center Portion (NIC) 4e3b3b
Administration 4e3b3b1
Computer Service Operations 4e3b3b2
People Service Operations 4e3b3b3
Net Interface (Station Agent and Net participation) 4e3b3b4
NIC Development 4e3b3b5
ARPA/RADC Project: Mini-Console 4e3b3c
Administration 4e3b3c1
System Development 4e3b3c2
ARPA/RADC Project: MPS Cooperation (Xerox) 4e3b3d

Administration 4e3b3d1

Modular Programming System Development (MPS) 4e3b3d2

ONR Project: System Developer's Intelligence System (SDIS) 4e3b3e

Administration 4e3b3e1

RADC Project: Baseline Management System Development Support 4e3b3f

Administration 4e3b3f1

SRI OVERHEAD ACTIVITIES 4e3b4

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

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

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

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

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

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

D. PODAC 4e4

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

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

We are convinced that unless we have a strong, constant, and pervasive attitude that we want to keep developing ourselves, and unless we consciously keep trying to do so, then we are fooling ourselves about seriously pioneering this augmentation system development. 4e4a1a

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. 4e4a1b

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

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

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

Each group meets weekly for two hours. 4e4b3

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

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

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

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

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

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

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

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

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

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

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

VI. REFERENCES 4f

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

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

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

(7404,) Proposal for Research [to RADC/ARPA ISU 71-94]. Network Information Center and Augmentation System Development. SRI No. ISU 71-94. Augmentation Research Center, Stanford Research Institute, Menlo Park, California 94025. 29 July 1971. Separately paged. (Entered in SRI-ARC Journal 18 October 1971.) 4f4

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

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

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

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

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

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

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

(9246,) Augmentation Research Center, Stanford Research

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

(9697,) D. C. Engelbart, W. K. English, J. F. Rulifson (SRI-ARC). Development of a Multidisplay, Time-Shared Computer Facility and Computer-Augmented Management-System Research. Augmentation Research Center, Stanford Research Institute, Menlo Park, California 94025. Rome Air Development Center, ARPA. AD 843 577. April 1968. 180p. 4f13

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

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

(10331,) D. C. Engelbart (SRI-ARC). To Launch FRAMAC. Augmentation Research Center, Stanford Research Institute, Menlo Park, California 94025. 4 May 1972. 3p. 4f16

(10457,) James C. Norton (SRI-ARC). Initial FRAMAC Meeting Notes. Augmentation Research Center, Stanford Research Institute, Menlo Park, California 94025. 23 May 1972. 14p. 4f17

(10459,) James C. Norton (SRI-ARC). Second FRAMAC Meeting Notes - Centering on the Concept of Frameworks. Augmentation Research Center, Stanford Research Institute, Menlo Park, California 94025. 1 June 1972. 39p. 4f18

(10478,) James G. Mitchell, A. J. Perlis, H. Van Zoeren. LCC: A Language for Conversational Computing. In: J. Reinfelds, M. Klerer, Eds. Interactive Systems for Experimental Applied Mathematics. Academic Press, New York. 1968. p.203-214. 4f19

(10479,) James G. Mitchell. LCC. In: Computer Science Research Review, Carnegie-Mellon University. 1969. p.21-29. 4f20

(10481,) L. A. Belady, M. M. Lehman. Programming System Dynamics -- Or the Meta-Dynamics of Systems in Maintenance and Growth. IBM Research Document RC 3546. T. J. Watson Research Center, International Business Machines Corporation, Yorktown Heights, New York. 17 September 1971. 4f21

(10553,) Richard W. Watson (SRI-ARC). FRAMAC Notes, May 19

[1972]. Augmentation Research Center, Stanford Research Institute, Menlo Park, California 94025. 19 May 1972. 2p. 4f22

(10703,) Augmentation Research Center, Stanford Research Institute, Menlo Park, California 94025. DNLS Preliminary Reference Guide. 21 June 1972. Separately paged. 4f23

(10869,) D. I. Andrews, H. G. Lehtman, W. H. Paxton (SRI-ARC). Tree Meta - A Metacompiler for the Augmentation Research Center. In process. Unpaged. 4f24

NETWORK INFORMATION CENTER

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

I. INTRODUCTION 5a

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

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

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

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

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

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

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

D. The Network Information Center (NIC) is an experiment in setting up and running a general purpose information service serving the ARPANET community (both those individuals and groups with direct access to the network, and those associated with work going on in the network but without direct access) with both online and offline services. The services offered and under development by the NIC have as their initial basic objectives: 5a4

1. 1) To help people with problems find the resources -- people, systems, and information -- available within the network community which meet their needs. 5a4a

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

II. THE NIC PUBLIC 5b

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

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

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

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

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

1. Students
2. Researchers
   university
   industry
   government
3. System Developers
   university
   industry
   government
4. Teachers
5. Managers
   university
   industry
   government
6. Computer Center Directors
7. Libraries and Other Information Services
8. The General Public
9. The Media

III. PRESENT NIC SERVICES 5c

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

1. Online: 5c1a

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

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

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

(b) Documents are now distributed by: 5c1a2b

i) placing the message or a link to the document in the receiver's "initial file". 5cla2b1

ii) sending hardcopy through the U.S. mail. 5cla2b2

Documents will shortly be distributed through the Network when sites have implemented the appropriate File Transfer Protocols 5cla2c

(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. 5cla2d

(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. 5cla2e

(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. 5cla2f

(3) Access to a number of online information bases through a special Locator file using NLS link mechanisms. 5cla3

(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 5cla3a

(b) Links to other files created by sites with information of potential Network-wide interest. 5cla3b

2. Offline: 5clb

(1) A Network Information Center Station set up at each site with: 5clb1

(a) A Station Agent to aid use of the NIC 5clb1a

(b) A Liaison to provide technical information about his site. 5clb1b

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

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

(a) Current Catalog of the NIC Collection 5c1b2a

(b) ARPA Network Resource Notebook 5c1b2b

(c) Directory of Network Participants 5c1b2c

(d) NIC User Guide 5c1b2d

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

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

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

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

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

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

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

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

1. The project has followed a research and development strategy of "bootstrapping", that is, of using the tools and techniques it has been developing in its own work,

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

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

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

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

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

V. OPERATIONS 5e

A. Computer Service Operations (CSO) 5e1

1. In the area of computer services, extensive measurement capabilities were added to the system to measure the efficiency of the TENEX operating system and NLS (#s,). A number of changes which appeared necessary as a result of these measurements were made and others are under study. 5e1a

2. Our hardware configuration contained a number of old, one-of-a-kind pieces of equipment brought over to the PDP-10 system from the previous XDS-940 system. These pieces of equipment have proven difficult to maintain and studies were launched on how to replace or upgrade this equipment. 5e1b

A new BBN network interface and a new DEC RP02 disc system were installed in the spring of 1972, replacing older unreliable equipment. 5e1b1

Hardware upgrading of our display system and its special core box has begun to provide temporary relief until a replacement system can be planned. 5e1b2

An additional 32K words of core has been added recently. 5e1b3

Studies leading to recommendations to add another channel, disc controller, and set of disc drives have been completed. These additions will provide more file storage capability and backup swapping capability. 5e1b4

The reliability improvements resulting from these measures and others under study should begin to be manifest in the summer of 1972. 5e1b5

3. Along with the above hardware improvements, improved practices and conventions have been evolved to handle new versions of software releases, both TENEX and NLS, and their checkout before being brought up for normal use. These conventions specify both frequency and time of day at which new systems can be brought up, and also specify documentation standards. 5e1c

4. One of the important aspects of CSO support has been

implementation, integration and maintenance of those programs necessary for communication with the ARPANET and hosts connected to it. The basic Network Control Program and TELNET Protocols are obtained as part of TENEX support from BBN. When we had a non-standard hardware interface to the network and during early protocol development, considerable effort was required in protocol implementation to create operating network programs. Less effort is now required, but this effort continues. We have also participated actively in working with the Network Working Group on protocol design and specification (proto qx,). 5e1d

B. People Services Operations (PSO) 5e2

1. During the past year ARC has developed several service functions that are now becoming operational for ARC users and NIC clientele. 5e2a

2. These functions (from activities such as RINS, NIC, Baseline Record, and Journal) and the forthcoming use of Deferred Execution (DEX) techniques have created new needs for people services support. 5e2b

3. As a result, we concentrated some of our effort on reorganizing these activities to allow more effective and efficient handling of routine and other tasks and to allow for easier expansion of the group size to meet needs of an increasing amount of throughput. The three aims were: 5e2c

To increase throughput to meet existing demands. 5e2c1

To become capable of expanding rapidly (in throughput quantity) to meet fluctuating service demands. 5e2c2

To work at minimizing costs while maximizing responsiveness to customer's needs and values. 5e2c3

4. This section describes in some detail the activities and tasks involved in setting up or running a PSO. We go into this detail because many people reading this report with a traditional computer service background may not appreciate the complexity of running an information service. Computer technology, while important, is not sufficient in and of itself to make possible such a

service. Such a service is only possible with a balanced set of computer tools, people support services, and the methodology, procedures, and training which meld them together into an effective higher level system. 5e2d

5. Therefore in order to create such a balanced system we launched a new approach to ARC's "people services operations". (see -- 7834,1a) 5e2e

The main thrusts were: 5e2e1

Organization
Physical Location and Configuration
Procedure Establishment and Documentation
Transcription Activities
Terminals
Personnel
Training

6. Organization 5e2f

A group with skills in handling paperwork and messages, and in using TNLS and DEX, was explicitly identified as PSO, and a group of advisors with skills in administration, documentation, and training was assigned to assist in getting PSO into formal operation. 5e2f1

7. Physical Location and Configuraion 5e2g

Office and workroom areas were expanded and relocated, to give the growing support operations more efficient location and arrangement. New tables, shelves, cabinets, and files were acquired and their arrangement worked out. 5e2g1

8. Procedure Establishment and Documentation 5e2h

Procedures were devised and documented for: 5e2h1

Use of TNLS (see -- 7470,) and DEX (see -- 9934,), 5e2h1a

The handling of transcription and other service requests. 5e2h1b

All related NIC activities -- clerical and secretarial. 5e2h1c

9. Transcription Activities 5e2i

Types of work to be handled: 5e2i1

Handwritten drafts
Tape recordings
Dictation notes
Offline documents
Online documents to be edited

Techniques for transcribing material into online files were developed: 5e2i2

Deferred Execution (DEX) covered at greater length in <C,DEX qx> 5e2i2a

This process makes use of terminal and magnetic tape recording equipment for initial input of data with actual entry into computer files deferred until periods of low system use (thereby resulting in less expensive use of the system for the processing of this work.) This system has been used to place online many documents of importance to the ARPANET community originally prepared offline. 5e2i2a1

Where and how long to store entered tapes for backup, the conventions for hierarchical statement entry, and when the transcriber should try to put hierarchical structure into documents are still under development. 5e2i2a2

TNLS 5e2i2b

TNLS is used largely for routine editing of online documents, and for entering high-priority items during off-peak load hours. 5e2i2b1

DNLS 5e2i2c

Display NLS is used for difficult editing of online documents and for some highly formatted documents. 5e2i2c1

Receiving processes 5e2i3

We set up a central receiving station. 5e2i3a

One person, with an alternate, handles users' questions regarding job status, time and cost estimates, etc. 5e2i3b

Priority determination process 5e2i4

A requestor specifies his preference for priority: 5e2i4a

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

Temporary storage of unassigned work 5e2i5

A log system using appropriate work request forms has been set up. 5e2i5a

We have a central storage place, organized for control of work by priority. 5e2i5b

Assignment process for transcription work 5e2i6

A work scheduler assigns incoming work to group members, balancing priority request with members' capabilities and workload. 5e2i6a

Later, priorities may be established by a bidding scheme. 5e2i6b

We plan to enlarge this effort to allow assignment to an outside pool of workers trained in DEX, both SRI people and contract manpower. 5e2i6c

Output processes 5e2i7

We have developed conventions for naming of temporary input files (special and separate for the catalog process) with provision for special instructions from the author. 5e2i7a

We have developed procedures for delivery of completed work to the requestor. 5e2i7b

10. Terminals 5e2j

We have made a thorough study of available teletype terminals and magnetic tape devices, and after experimental use of several, have leased nine TI terminals and six Termicettes, for use with DEX. 5e2j1

11. Personnel 5e2k

We have added several new staff members with contributions to make to NIC. Two writers who can also teach were active in PSO development. Three new staff members were added to the document preparation, transcription and distribution efforts. 5e2k1

12. Training 5e2l

Classes in TNLS and DEX were held for ARC and network people. Manuals were prepared. A more detailed discussion of training is given later, see--,5g10). 5e2l1

13. A detailed list of the types of tasks this PSO group and associated information handling people perform to support the NIC is given below because it is important for people to understand the range of activities that are required even with automated aids to support a service such as the NIC. 5e2m

PLANNING AND SCHEDULING 5e2m1

Goal setting 5e2m1a
Service design 5e2m1b
Site Station aid planning 5e2m1b1
Functional document design 5e2m1b2
NIC Collection design 5e2m1b3
Station collection plan 5e2m1b4
Reference service design 5e2m1b5
Catalog design 5e2m1c
Procedure establishment 5e2m1d
Discussion 5e2m1d1
Procedure writing 5e2m1d2
Experimentation 5e2m1d3
NIC facility design 5e2m1e
Work flow scheduling 5e2m1f
NIC time and cost studies 5e2m1g

GENERAL SUPPORT 5e2m2

Dictation	5e2m2a
Phone	5e2m2b
Orders and financial records	5e2m2c
Timecards	5e2m2d
Visitor arrangements	5e2m2e
NIC travel arrangements	5e2m2f
NIC facility upkeep	5e2m2g
STATION PHONE ACTIVITY	5e2m3
Station phone answer	5e2m3a
NIC outgoing calls	5e2m3b
MAIL SINGLE NIC PIECES	5e2m4
Incoming mail processing	5e2m4a
Single mailings	5e2m4b
ACQUISITION OF NETWORK INFORMATION	5e2m5
Network resources	5e2m5a
Network personnel	5e2m5b
Network publication references	5e2m5c
CHOICE, ANALYSIS, OF INFORMATION	5e2m6
Analysis for bulletins	5e2m6a
Analysis for functional documents	5e2m6b
Selection of publications	5e2m6c
Abstracting	5e2m6d
ACQUISITION OF PUBLICATIONS	5e2m7
Checking holdings	5e2m7a
Order form preparation	5e2m7b
Receipt, record changing	5e2m7c
OFFLINE CATALOGING WORK	5e2m8
Coding	5e2m8a
Checking of coding, revision	5e2m8b
Proofing and revision	5e2m8c
Recoding of old material	5e2m8d
Catalog offline records	5e2m8e
Old catalog offline work	5e2m8f

FILE-BUILDING ONLINE 5e2m9

Input of new citations 5e2m9a
Input of old citations 5e2m9b
Editing of new citations 5e2m9c
Editing of old citations 5e2m9d
Bulletin creation 5e2m9e
Bulletin editing 5e2m9f
Catalog creation 5e2m9g
Catalog editing 5e2m9h
Catalog file manipulation 5e2m9i
Functional documents input 5e2m9j
Mailing list input 5e2m9k
Letter online input 5e2m9l
Other online text input 5e2m9m
Other text input, DEX 5e2m9n
Identfile maintenance 5e2m9o

PHYSICAL PROCESSING 5e2m10

Readying of Journal printout 5e2m10a
Readying of other work 5e2m10b
Collating 5e2m10c
Stamping, Punching 5e2m10d
Xeroxing of documents 5e2m10e
Line printer output 5e2m10f
Outside repro contact 5e2m10g

DISTRIBUTION 5e2m11

Mailing list maintenance 5e2m11a
Labels, envelope preparation 5e2m11b
Pickup and delivery 5e2m11c

STORAGE AND MAINTENANCE 5e2m12

NIC Master collection 5e2m12a
Extra copies 5e2m12b
Supplies 5e2m12c

VISUAL AIDS 5e2m13

Chartmaking 5e2m13a

TRAINING 5e2m14

Instruction 5e2m14a
Development of training aids 5e2m14b

REFERENCE WORK 5e2m15

Locating citations for Net 5e2m15a
Locating documents for Net 5e2m15b
Literature search 5e2m15c

C. Let us now look at each of the services provided and see what has been involved in making them available beyond the changes described above, why they were made available, and some future plans. 5e3

D. There are two major areas of changes to ARC caused by providing NIC services that deserve mentioning: planning and providing more reliable and efficient computer services, and planning and providing more varied and extensive clerical and other services provided by and for people. 5e4

VI. ONLINE SERVICES 5f

A. ACCESS TO NLS 5f1

1. The ARC ONLine System (NLS) is an evolving system which we view as an integrated set of tools for doing general intellectual work (,4d1,). To this end NLS has, at this time, powerful document creation, editing, production, and studying capabilities, dialog support functions for online communication both simultaneous and distributed in time, bibliographic catalog-making capabilities, programming aids and facilities, some basic information retrieval abilities, and some, as yet, rudimentary management and other planning aids. 5f1a

2. A subset of these capabilities, felt to be of prime value to initial NIC use centered around document creation, editing, production, and studying as well as dialog support, has been thoroughly documented for NIC clientele. 5f1b

3. We knew that most systems on the network supported typewriter terminals rather than displays so that during the conversion from the XDS-940 to the PDP-10, a typewriter version of the system was designed and implemented --,4d1a3a). 5f1c

In thinking about the problems which could exist in supporting all the varieties of typewriter terminals on the network, we (to keep ARC's thoughts clearly separate from the net's) felt that it would be better to have most of these differences handled by a standard network protocol. Therefore, we workgd actively with the Network Working Group (NWG) in establishing a network virtual terminal protocol (TELNET) see--,6b1). This protocol has succeeded in allowing access to TNLS from different systems and terminals. 5f1c1

4. NLS, as it has historically developed, is oriented in its command language design for expert users. 5f1d

This orientation toward highly trained, experienced users is not completely suitable for the clientele of the NIC, comprised of some who use the system often enough to become experts and others who want to use the system infrequently. Therefore, thought has recently gone into studying what changes are needed in the NLS command language syntax to provide a range of modes from novice to expert, and what additional help and tutorial capabilities need to be built into the system. These changes will be implemented in the coming months. 5f1d1

5. At the present time anyone with access to a typewriter terminal connected to the network and with an entry in our identification file ,see--,4a4) (entries can be made in this file directly by network users) can have access to NLS. 5f1e

We generally find between 1 and 3 users from the network using TNLS during prime hours. The highest number observed has been 7 simultaneous network users. The number of logins a day from the network has been averaging around 40 - 50, with a variation between 30 and over 100. We expect the number to increase significantly both as the network grows and as our hardware reliability improves, as discussed earlier. 5f1e1

6. A system for allowing access to the display version of NLS (DNLS) from the network using IMLAC display terminals equipped with a keyset and mouse has been developed

jointly with the Xerox Palo Alto Research Center and tested with users from UCLA-NMC and BBN-TENEX. 5f1f

We expect to continue experimental use of DNLS over the network and eventually to offer DNLS as a regular service. We are currently studying how to provide DNLS service from low-cost alpha-numeric displays equipped with keyset and mouse. 5f1f1

7. Documents are presently created by a user at a keyboard device connected to TNLS via the network. We are working to allow entry of documents into NLS which were initially prepared in other host computers. At least one site, MIT-DMCG, has been entering documents in NLS by preparing locally a file of NLS commands and document text and transmitting it into NLS as a simulated teletype. The schemes under study will simplify this process. 5f1g

B. AIDS TO COLLABORATION 5f2

1. We envision a wide variety of collaboration aids to help geographically distributed people work closely together. One such system being developed and offered as a NIC service is the Dialog Support System (DSS). The first steps in the creation of a DSS have been taken in the implementation of Journal, Number, and Identification systems. 5f2a

2. As discussed in more detail above, the Journal is a system for capturing recorded dialog items (in the form of documents and messages) and for distributing these items online, offline, and through the network to the appropriate recipients. 5f2b

3. When an item is submitted to the Journal, a unique number is associated with it, either obtained at the time of submission or previously from the Number system. This unique number is used for cataloging purposes and as the name of the item for later reference and retrieval. 5f2c

4. Once submitted, the items become read-only; statements in a Journal item can be uniquely and precisely referenced in future documents with assurance that the reference will remain meaningful. 5f2d

5. At the time of submission, or any time later, documents can be distributed to one or more individuals, either singly or as members of groups by indicating to the system a list of unique identifications called IDENTS. 5f2e

New identifications can be created at the time of submission or at other times by use of the Identification system. 5f2e1

6. The IDENTS are usually a person's or group's initials. The IDENTs are automatically assigned by the Identification System when a person's or group's name is entered into an identification file by use of the system. 5f2f

When one is sending an item to a group, one need only use the group's IDENT and the system will deliver to the the membership of the group. One can also indicate distribution to only the coordinator of the group. 5f2f1

7. If one does not remember a person's or group's IDENT at the time of submission, a query capability allows it to be retrieved. 5f2g

8. The Identification System has provisions for collecting other relevant information such as a phone number, network site affiliation, and preferred method of document delivery (online as a citation in the receiver's Initial File, offline by hardcopy through the mail, or both). 5f2h

9. The information in the identification file is used by the Journal System during document submission and delivery. The information in the identification file is also used to automatically prepare directories of individuals and dialog groups as described later. 5f2i

SAMPLE MESSAGE SENDING SESSION 5f2j

The following is a demonstration of how a message is submitted to the Journal by a Network user (including login, NLS access, and logout procedure). Material in square brackets is fed back by the system. Material in parentheses is commentary. The symbols *, @, &, && are system heralds and are not shown in brackets. 5f2k

```
@Login SP DOE SP DDD SP 1 CR          (A user named DOE
   logs in to the system - his ID is DDD)                  5f2l

[JOB 11 ON TTY14  3-AUG-71  17:11]                         5f2m

@nls CR                               (The user accesses
   the NLS system)                                         5f2n
```