ANALYSIS OF COMPUTER USAGE AND COSTS 1

SUMMARY 2

INTRODUCTION 2a

At the request of ARPA we have conducted a utilization study and a cost analysis of our computer operations to assess the degree of utilization of our facilities and to determine the relative costs of the various services we are providing to our community of users. 2a1

The results presented here are our best estimates of what our real situation is and they may obviously vary with our operating conditions. However, the data we have collected over the past 9 months are indicating a reasonable stability of our operating environment. Consequently, under present conditions, our data should represent fairly well the real situation within our installation. 2a2

CPU TIME 2b

For February 73, the average distribution of the CPU time was the following (see figure 2): 2b1

PERCENTAGE OF TOTAL CPU TIME	FROM 10 AM-11 AM	FROM 8 AM-5 PM	FROM 5 AM-8 AM
Charged to Users	56.3	57.1	24.0
Idle Time	6.3	7.6	47.0
System Overhead	37.4	35.3	29.0

2b2

Thus, of the total CPU time available, only a maximum of approximately 60% is being charged to user accounts. We refer to that fraction as the "CPU time charged to users" or "the user CPU time". Hence, if a group of users is using 10% of the user CPU time it is in fact getting only 6% of the total CPU time available. 2b3

We estimate that presently each week 40 hours of CPU time are being charged to user accounts. It can be approximately broken down as follows: 2b4

Working Days	5 am to 8 am . . .	3 HRS
Working Days	8 am to 5 pm . . .	23 HRS
Evenings	5 pm to 10 pm . . .	10 HRS
Weekend	. . . .	4 HRS

	--
Total	40 HRS

2b4a

CONNECT TIME 2c

The corresponding connect time charged to user accounts is approximately 1,000 hours per week. Except for one week where heavy processing was done during the night period, this total connect time varies from 800 hours to 1150 hours. Thus it is fairly stable. 2c1

The average ratio of user CPU time over connect time is approximately .04. This indicates that, on the average, each user is getting 4% of the computing capacity when working on the system. This is indeed the case for DNLS users. However, for TNLS users the ratio is closer to .02, and for some background systems the ratio is usually around .06. 2c2

NUMBER OF USERS 2d

From this ratio it follows that since the maximum CPU time charged to users is 60 %, the maximum number of users who can simultaneously utilize the system without significantly overloading it is the following. 2d1

USER TYPE	MAXIMUM NUMBER OF USERS FROM ONE CATEGORY ONLY
TNLS USERS	30
DNLS USERS	15
HEAVY TNLS USERS	20
HEAVY DNLS USERS	10

2d1a

Our present experience indeed confirms these numbers; usually we have from 8 to 10 DNLS users and around 10 TNLS users on the system during daytime. 2d2

COST OF COMPUTING OPERATIONS 2e

We have calculated the total costs of our computing operations in two ways, namely, by including all current costs only - the low estimate - and then by adding the amortized costs of ARC purchased equipment in current use - the high estimate -. Both numbers include the costs for hardware (including terminal costs), operation and maintenance costs, personnel costs, and the SRI overhead. Our estimated present facility operation costs are the average between both high and low costs. 2e1

They can be summarized as follows:

COST PER:	ESTIMATED	LOW ESTIMATE	HIGH ESTIMATE	2e2
YEAR	$750,000	$660,000	$840,000	
MONTH	$62,500	$55,000	$70,000	
WEEK (1)	$15,000	$13,200	$16,800	2e3

COSTS ALLOCATED TO CPU TIME ONLY

COST PER:	ESTIMATED	LOW ESTIMATE	HIGH ESTIMATE	
HOUR OF CPU TIME (2)	$375	$330	420	
MINUTE OF CPU TIME (2)	$6.25	$5.50	$7	2e4

COSTS ALLOCATED TO CONNECT TIME ONLY

COST PER:	ESTIMATED	LOW ESTIMATE	HIGH ESTIMATE	
HOUR OF CONNECT TIME (3)				
Average Cost	$15	$13.2	$16.8	
If DNLS Users Only	$20	$17.2	$22.4	
If TNLS Users Only	$10	$ 8.6	$11.2	2e5

NOTES

(1) a year of 50 weeks is assumed;
(2) based on 40 hours of CPU time per week (CPU time charged to user accounts only);
(3) based on an average of 1,000 hours of connect time per week (assuming costs allocated to connect time only); 2e5a

UTILIZATION OF COMPUTING FACILITIES 2f

As far as the utilization of our computing facilities is concerned the situation can be summarized as follows: 2f1

1) During our prime time, i.e.,from 8 a.m. to 5 p.m., we are operating at capacity (see figures 3 & 4). In order to preserve acceptable response time, we are now controlling computer access by limiting the number of people allowed to log in during that time. We plan to replace this simple control procedure by a group allocation scheme which is presently being developed (13227,). 2f2

2) The period from 5 a.m. to 8 a.m. could be utilized much more extensively. Presently, the average idle time during that period is still around 45% (figure 4). East Coast Network users could be our prime clients for that time. 2f3

3) The evening period is presently filling up very rapidly (see figure 4), and if the recent trend continues we will operate in the near future at full capacity during that period. 2f4

4) The weekend remains the only time period where plenty of spare capacity could still be taken advantage of. 2f5

5) DEC maintenance is done from 10 PM to 3 AM twice per week and the system dump is done daily from 3AM to 5AM. 2f6

USAGE PROFILE 3

BREAKDOWN OF CPU TIME CHARGED TO USER ACCOUNTS 3a

SUMMARY 3a1

On the average, approximately 40 hours of CPU time are presently being charged every week to user accounts. See table (2b4a,). This time can be broken down as follows: 3a1a

Working Days	5 am to 8 am . . .	3 HRS
Working Days	8 am to 5pm . . .	23 HRS
Evenings	5 pm to 10 pm . .	10 HRS
Weekend	. . . .	4 HRS
		--
	Total	40 HRS

3a1a1

This is the "charge time" to which costs will have to be allocated in some fashion. We estimate that under ideal conditions the maximum amount of CPU time which will be charged to user accounts will be the following. 3a1b

Working Days	5 am to 8 am . . .	6 HRS
Working Days	8 am to 5pm . . .	25 HRS
Evenings	5 pm to 10 pm . .	20 HRS
Weekend	. . . .	9 HRS
		--
	Total	60 HRS

3a1b1

In the following tables we summarize some recent measurements of our operations on which the preceding conclusions are based. 3a1c

OVERALL WEEKLY CHARGES TO ALL USER ACCOUNTS 3a2

(This includes everything, i.e., the CPU time charged during the day, at night, and during the weekend) 3a2a

3a2b

WEEK OF	CPU TIME (Hours)	CON. TIME (Hours)	RATIO CPU/CON
Jan 14 - Jan 20	42.0	924.8	.045
Jan 21 - Jan 27	44.0	1152.9	.039
Jan 23 - Feb 3(*)	66.0	1553.5	.042
Feb 4 - Feb 10	41.0	988.5	.041
Feb 11 - Feb 17	32.9	806.8	.041
Feb 18 - Feb 24	36.8	787.8	.047
Feb 25 - Mar 3	39.4	1106.0	.036

AVERAGE	43.1	1045.2	.041	3a2c

(*) During the week of January 23 heavy processing was done at night and during the weekend (the catalog and documentation). 3a2d

BREAKDOWN BY USER GROUPS 3a3

An analysis of CPU time usage by various user groups shows how and by whom our computing power is being utilized. The following table (week of February 5) illustrates fairly well the distribution of the user CPU time among the different types of users. Because we want to illustrate the prevailing conditions during working days, only the CPU time charged during prime time,i.e., from 5 AM to 5 PM, is accounted for. It should be noted that during that particular week the computer was down over that period for about 16 hours which accounts for the low overall CPU time which was charged. Otherwise, the numbers are characteristic of our operations. 3a3a

USER GROUP	NUMBER	CPU TIME	CON. TIME	RATIO	% OF	
TOTAL						
	of USERS	(Hours)	(Hours)	CPU/CON	CPU TIME CHARGED TO USERS	
	(1)	(2)	(3)	(4)	(5)	3a3b
STAFF	8	2.7	71.3	.038	11.7	
PSO	5	2.8	56.1	.050	13.7	
NLS PROGRAMMERS	15	4.2	121.9	.034	20.5	
TENEX PROGRAMMERS	4	2.9	45.6	.064	14.1	
NIC STAFF	7	1.6	52.1	.031	7.8	
RADC VIA NETWORK	10	1.2	65.5	.044	5.9	
OTHER NET USERS	25 (6)	3.3	185.1	.018	16.1	
OVERHEAD USERS (7)	5	2.1	38.2	.055	10.2	
	--	----	-----		-----	
TOTAL	79	20.5	638.5		100.0	3a3c

COMMENTS: 3a3c1

1) Number of users in the group who logged in at least once during the period considered. 3a3c2

2) Only the CPU time charged to user accounts; roughly 50 % of all CPU time available. 3a3c3

3) Total terminal connect time. 3a3c4

4) Average ratio of CPU time over connect time. See discussion below. 3a3c5

5) Again, this is not the percentage of total CPU time available, but the relative percentages of total CPU time charged to users. 3a3c6

6) Most network users are logging in under the name of their installation, i.e., under names such as MITRE-TIP, UCSB, UCLA-NMC and others, and, therefore, the actual number of network users per week is much greater than the 25 indicated above. 3a3c7

7) Includes system users such as the printer, documentation, the catalog, system, background, and operator. 3a3c8

BREAKDOWN OF GROUPS BY ARC IDENTS 3a3d

STAFF (Other than below)
DCE, JCN, RWW, PR, DVN, MFA, BAH, MEH 3a3d1

PSO (Clerical Support)
KFB, LLL, KIRK, MEJ, NDM 3a3d2

NLS PROGRAMMERS
WLB, CFD, JDH, CHI, DSK, HGL, EKM, JFV, PARC(XEROX) 3a3d3

TENEX PROGRAMMERS
KEV, DCW, WRF, DIA 3a3d4

NIC
JEW, JBN, EJF, MDK, SRL 3a3d5

OVERHEAD USERS
BACKGROUND, CAT, PRINTER, OPERATOR, PETERS,
DOCUMENTATION, SYSTEM 3a3d6

RADC 3a3d7

NETWORK USERS 3a3d8

DATA FROM THE WEEK OF FEBRUARY 19 3a4

(from 5 a.m. to 5 p.m. for working days only)
(the 19th was a holiday) 3a4a

USER GROUP	NUMBER	CPU TIME	CON. TIME	RATIO	% OF

TOTAL						
	of USERS	(Hours)	(Hours)	CPU/CON	CPU TIME CHARGED TO USERS	
	(1)	(2)	(3)	(4)	(5)	3a4b
STAFF	8	4.3	110.4	.039	14.7	
PSO	5	5.8	71.6	.081	19.7	
NLS PROGRAMMERS	14	4.6	105.0	.044	15.7	
TENEX PROGRAMMERS	4	1.9	44.0	.043	6.4	
NIC	5	1.2	45.2	.026	4.0	
RADC	14	1.2	69.8	.017	4.0	
NETWORK USERS (6)	25	2.9	108.9	.026	9.8	
OVERHEAD USERS (7)	7	7.6	79.0	.097	25.7	
	--	----	-----		----	
TOTAL	82	29.5	633.9		100.0	3a4c

RATIO OF CPU TIME OVER CONNECT TIME 3b

This ratio tells us the average percentage of CPU time the average user is getting when he is on the system. This average varies from user to user, with the nature of the work being performed, and with the time of the day. (See tables below). 3b1

An average ratio of .04 indicates that the average user is getting, on the average, 4% of the CPU time when he is on the system. Thus, if the average overhead is 60% of the available CPU time, then, on the average, our system cannot accommodate simultaneously more than 15 such users. However, TNLS users have a ratio near .02, whereas experienced DNLS users have all a ratio near .04. Hence, it is the mix of both categories of users, combined with the fact that during peak utilization periods all averages are somewhat lower, that allows us presently to accomodate during the day around 20 userss. 3b2

To summarize, assuming that the total overhead is 60 %, our system can accommodate on the average

- 15 users with a utilization ratio of .04 (DNLS users)
- 20 users with a utilization ratio of .03
- 30 users with a utilization ratio of .02 3b3

NUMBER OF USERS 3c

See figures 7 and 8 for the average distribution of both the total number of users and the average number of network users. 3c1

COMPUTER USAGE ANALYSIS 4

BREAKDOWN OF CPU TIME USAGE 4a

MONTHLY AVERAGES OF CPU TIME USAGE BETWEEN 8 AM. AND 5 PM. 4a1

	MONTH					
% CPU TIME	OCT	NOV	DEC	JAN	FEB	4a1a
Charged to Users	57.6	63.5	57.0	56.4	57.1	
Idle Time	13.6	8.3	10.3	10.0	7.6	
Overhead	27.9	27.2	31.7	32.5	34.3	4a1b
BREAKDOWN OF OVERHEAD						4a1c
I/O Wait	8.9	7.7	11.2	11.6	12.0	
Scheduler	8.7	10.1	9.0	9.7	8.7	
Clock	3.0	2.8	4.6	5.3	7.3	
Garbage Collection	2.4	2.8	2.7	3.0	2.9	
Teletype Simulation	4.9	3.8	4.2	2.9	4.4	
Total Overhead	27.9	27.2	31.7	32.5	34.3	4a1d

BREAKDOWN BY SUBSYSTEMS. 4b

The following tables show in what subsystems the CPU time charged to users is being spent. We have grouped these subsystems into 6 categories which correspond roughly to the various types of activities going on within ARC. They show clearly how, on the average, our computing capacity is being allocated. 4b1

PERCENTAGES OF TIME SPENT IN VARIOUS SUBSYSTEMS 4b2

	OCT	NOV	DEC	JAN	JAN	JAN	
PERIOD	24HRS	24HRS	24HRS	24HRS	8-5	10-11	4b2a
NLS	40.7	49.2	47.7	43.8	51.9	56.1	
HARD COPY	14.5	15.8	15.2	23.0	17.1	17.1	
JOURNAL	2.2	3.1	3.0	2.4	.6	1.0	
SUPPORT	18.3	19.5	19.9	19.2	19.0	17.5	
SYST. SUPPORT	11.1	8.8	12.3	7.7	7.4	4.5	4b2b

SUBSYSTEMS IN THESE CATEGORIES 4b3

(The numbers in parentheses following the names of the subsystems are the percentages of time charged to users in these subsystems in January, during the 8 a.m. to 5 p.m. period of the working days) 4b3a

NLS (51.9 %) 4b3b

DNLS (31.3), TNLS (12.0 %), NTNLS (5.8 %)(se figure 9), NDNLS (.8 %), NLSL10, DEX, CASSET 4b3b1

HARDCOPY OUTPUT (17.1 %) 4b3c

OUTPRC (5.5 %), SYSJOB (11.6%). S360, MTACPY, XOUPRC 4b3c1

JOURNAL (.6 %) 4b3d

JOURNL, OLJDEL, RECOVF, SLINKR 4b3d1

(It should be noted here that the journal background systems are mostly run during the nightime) 4b3d1a

SYSTEM SUPPORT (19 %) 4b3e

EXEC (15.6%), BSYS (.9%), SUPERW (1%), DELD, BIGBRO, ACCSRI, NOTIFY, DUMPER, FAIL, FTP, SNDMSG, TTYTST, SRCCOM, QUERY 4b3e1

SYSTEMS DEVELOPMENT (7.4 %) 4b3f

PRIV (3.5%), L10 (1%), TENLDR (1%), PRVPRC(.7%), TECO (.5%), LOADER, LTBNLS, UTILITY, SDDT, BASIC, MACRO, RUNFIL, SMFS 4b3f1

ARC FACILITY COSTS 5

Present Configuration 3/7/73 (TOTAL FACILITY: $ 43,592 /mo.) - includes $ 30,065 monthly costs plus $ 13,527 monthly "amortized costs" for equipment in use that is fully paid for. 5a

LEASED DEC EQUIPMENT: 5b

BASIC FACILITY leased from DEC (*) .	$ 13,986 /mo	
DEC DISK PACK EQUIPMENT (**) . . .	$ 6,514 /mo	
DEC ME10 Memory (16k) Addition . .	$ 1,250 /mo	
DEC Maintenance	$ 5,410 /mo	

TOTAL . . .	$ 27,160 /mo	5b1

NOTES: 5b2

(*) Includes: 5b2a

KA10 Arithmetic Processor 5b2a1
KM10 Fast Register 5b2a2
KT10A Dual Mem Protect Relocate 5b2a3
TM10A Mag Tape Control 5b2a4
TD10 DECTape Control 5b2a5
DC10A Data Line Scanner Control 5b2a6
TU30-B 7-Channel Mag Tape (two) 5b2a7
TU55 DECTape Transport (two) 5b2a8
DC10B 8-Line Group Unit 5b2a9
MA10 Core Memory (eight) 5b2a10
MC10 Memory Ports (24) 5b2a11

(**) Includes: 5b2b

DF10 Data Channel (two)
RPO2 Disk Controller (two)
RPO2 Disk (six) 5b2b1

OTHER LEASED EQUIPMENT: 5c

Dataphones (1)	$ 245 /mo	
Couplers (2)	$ 120 /mo	
Cassette Recorders (3)	$ 640 /mo	
T-I Terminals (4)	$ 900 /mo	

TOTAL	$ 1,905 /mo	5c1

NOTES: 5c2

(1) 7 leased on the IPT contract: cost is about $ 245 /mo. 5c2a

(2) 8 leased on the IPT contract: cost is about $ 120 /mo. 5c2b

(3) 6 leased on the IPT contract: cost is about $ 640 /mo. 5c2c

(4) 9 leased terminals: cost is about $ 900 /mo. 5c2d

ARC-PURCHASED EQUIPMENT IN CURRENT USE: 5d

(The total cost is : $421,090; amortized @ 40 mo = $ 10,527/mo) 5d1

A/D Converter (1)	$ 6,500	
Data Products Line Printer (2)	$ 50,000	
Display Control System (3)	$ 16,000	
Display Consoles and TV System (4)	$ 55,000	
Tasker Display System (5)	$ 118,000	
Execuport Terminals (6)	$ 7,800	
External Core (Xcore) (7).	$ 32,240	
I/O Control Box (8)	$ 12,700	
Input Devices Controller (IDC) (9)	$ 9,800	
Paging Box (BB&N) (10)	$ 54,000	
Other equipment (11)	$ 20,000	
Real Time Clock (12)	$ 1,250	
T-I Terminals (13)	$ 20,000	
TTY Patch Panel (14)	$ 4,000	
Xcore Interface Box (15)	$ 13,800	
TOTAL	$ 421,090	5d2

NOTES: 5d3

(1) Purchased on IPT contract: cost was about $ 6,500 5d3a

(2) Leased, then GFE: cost was about $ 50,000 5d3b

(3) SRI Constructed: cost about $ 16,000 5d3c

(4) 12 (plus spares) were purchased on IPT contract: cost was about $ 55,000 5d3d

(5) Purchased on IPT contract: cost was about $ 118,000 5d3e

(6) 2 purchased, total $ 7,800 5d3f

(7) SRI Constructed: cost was about $ 32,240. 5d3g

(8) SRI Constructed: cost was about $ 12,700 5d3h

(9) SRI Constructed: cost was about $ 9,800 5d3i

(10) Purchased on IPT contract: cost was $ 54,000 5d3j

(11) Such as amplifiers, racks, video distribution cabling, etc. A rough estimate of the cost is $ 20,000 5d3k

(12) Purchased on IPT contract: cost was about $ 1,250 5d3l

(13) 8 purchased , cost approximately $ 20,000 5d3m

(14) SRI Constructed: cost was about $ 4,000 5d3n

(15) SRI Constructed: cost was about $ 13,800 5d3o

IPT SUPPLIED EQUIPMENT 5e

(The total cost is: $ 120,000; amortized @ 40 mo = $ 3,000/mo) 5e1

Bryant Drum (1)	$ 120,000	5e2
IMP and IMP Interface (2)	$ 80,000	5e3

NOTES: 5e4

(1) Supplied by ARPA directly for IPT contract: estimated cost to ARPA $ 120,000. Considered a part of the ARC Facility costs for this estimate. 5e4a

(2) The Interface Message Processor and IMP Interface (IMP) were supplied by ARPA directly for IPT contract: estimated cost to ARPA $ 800,000. NOT considered a part of the ARC Facility costs for this estimate. 5e4b

OPERATING MAINTENANCE SUPPLIES AND EQUIPMENT 5f

The total cost is estimated to be $ 1,000 /mo 5f1

ARC PERSONNEL COSTS 6

General: 34 people in ARC + 1 "SRI borrowed" 6a

MONTHLY PROJECT COSTS: TOTAL: $ 80,000 /mo.; see (loadfactor) 6b

ARPA/IPT Project Supported 6c

(TOTAL: $ 67,100 /mo.) 6c1

Operations 6c2

Hardware	people: 3.5	$ 10,000 /mo (core facility)	
Software	people: 3.5	$ 10,000 /mo (core facility)	
User Int	people: 2.5	$ 7,100 /mo (core facility)	
Administ	people: .5	$ 1,500 /mo (core facility)	
Clerical	people: 2.0	$ 5,600 /mo	
Total	. . . .	$ 34,200 /mo	6c2a

[Average ARC rates used for rough costs per category] 6c2b

Development 6c3

Software	people: 5.0	$ 14,300 /mo	
Other	people: 1.0	$ 2,900 /mo	
Total		$ 17,200 /mo	6c3a

NIC 6c4

Software	people: 1.0	$ 2,900 /mo	
Clerical	people: 2.0	$ 5,700 /mo	
Other	people: 2.5	$ 7,100 /mo	
Total		$ 15,700 /mo	6c4a

ONR Project Supported 6d

people: 1.1 $ 3,200 /mo 6d1

RADC Project Supported 6e

people: 3.4 $ 9,700 /mo 6e1

SRI OVERHEAD 6f

(ARC overhead costs not directly applicable to cost-per-unit of NLS service except as a factor reducing the direct ARC project personnel charges on ARC projects - ARC overhead costs are included in SRI overhead rates, applied to project salary costs and included in the above project personnel cost figures) 6f1

- people: 7.0 6f2

CORE COMPUTER FACILITY PERSONNEL COSTS BASED ON THE ABOVE COSTS 6g

Hardware	people:	3.5	($ 10,000 /mo)	
Software	people:	3.5	($ 10,000 /mo)	
User Int	people:	1.5	($ 4,300 /mo)	
		----	---------	
Total	people:	10.0	($ 24,300 /mo)	6g1

CORE COMPUTER FACILITY COSTS BASED ON THE ABOVE COSTS 7

Basis I: 7a

Hardware (current costs only):	$ 30,967 /mo incl 3% SRI fee	
Personnel (core costs only):	$ 24,300 /mo	

Total	($ 55,267 /mo)	7a1

Basis II: 7b

Hardware (with amort costs):	$ 44,900 /mo incl 3% SRI fee	
Personnel (core costs only):	$ 24,300 /mo	

Total	($ 69,200 /mo)	7b1

Comments: 7c

In addition to the core computer facility personnel costs included above, the rest of ARC, the NIC, and other project support contribute to new features, much additional user interaction, etc. How to factor these costs in (if at all) remains to be worked out. 7c1

NOTES 8

(loadfactor) The calculation of the ARC project personnel cost per month is as follows: 8a

From SRI salary records - ARC Monthly salary after about 10% for sick leave and vacation is subtracted is: $ 37,000 8a1

Of this figure, only about 78% (or $ 28,800 /mo) is charged directly to projects. 8a2

The balance is charged to SRI overhead for meetings, courses. some conferences, overall management of ARC, some travel, and other non-direct activities. This is admittedly somewhat arbitrary in the case of ARC, but is consistent with the proportion of such charges experienced by other SRI organizations. 8a2a

To the above project salary costs (or $ 28,800 /mo) are applied three factors, reulting in a rate of 2.76 these days. 8a3

They are Payroll Burden, SRI overhead, and Fee. 8a3a

This results in total ARC project personnel charges (loaded) of about $ 80,000 /mo. 8a4

The above facility costs are intended to be just for the facility-cost estimate support. The interconnectedness of the development, operations, NIC, and other ARC projects makes it difficult to realdy figure the true cost of NLS service to the "average" non-ARC user. and certainly, the cost to ARC users is different from that. 8b

BRIEF FACILITY CONFIGURATION LISTING 9

- LEASED DEC EQUIPMENT 9a
  - BASIC FACILITY leased from DEC 9a1
    - KA10 Arithmetic Processor 9a1a
    - KM10 Fast Register 9a1b
    - KT10A Dual Mem Protect Relocate 9a1c
    - TM10A Mag Tape Control 9a1d
    - TD10 DECTape Control 9a1e
    - DC10A Data Line Scanner Control 9a1f
    - TU30-B 7-Channel Mag Tape (2) 9a1g
    - TU55 DECTape Transport (2) 9a1h
    - DC10B 8-Line Group Unit 9a1i
    - MA10 Core Memory (8) 9a1j
    - MC10 Memory Ports (24) 9a1k
  - DEC DISK PACK EQUIPMENT 9a2
    - DF10 Data Channel (2) 9a2a
    - RPO2 Disk Controller (2) 9a2b
    - RP02 Disk (6) 9a2c
  - DEC ME10 Memory (16k) Addition 9a3
  - DEC Maintenance 9a4
- OTHER LEASED EQUIPMENT: 9b
  - Dataphones (7) 9b1
  - Couplers (8) 9b2
  - Cassette Recorders (6) 9b3
  - T-I Terminals (9) 9b4
- ARC-PURCHASED EQUIPMENT IN CURRENT USE: 9c
  - A/D Converter 9c1
  - Data Products Line Printer 9c2
  - Display Control System 9c3
  - Display Consoles (12, plus spares) and TV System 9c4
  - Tasker Display System 9c5
  - Execuport Terminals (2) 9c6
  - External Core (Xcore) 9c7
  - I/O Control Box 9c8
  - Input Devices Controller (IDC) 9c9
  - Paging Box (BBSN) 9c10
  - Other equipment, such as amplifiers, racks, video distribution cabling, etc. 9c11
  - Real Time Clock 9c12
  - T-I Terminals (8) 9c13
  - TTY Patch Panel 9c14
  - Xcore Interface Box 9c15
- IPT SUPPLIED EQUIPMENT: 9d
  - Bryant Drum 9d1
  - [Interface Message Processor (IMP) and IMP Interface] 9d2

LIST OF FIGURES 10

(J15066) 13-MAR-73 11:59; Title: Author(s): Rech, Paul , Norton, James C. /PR JCN ; Distribution: /sri-arc dls rek2 lgr ; Sub-Collections: SRI-ARC; Clerk: PR;
Origin: <RECH>COSTANA.NLS;11, 13-MAR-73 11:34 PR ;

HGL 13-MAR-73 15:28 15067

Response to CIRAD Request for Comments on Their Report

This will be sent with Xeroxed copies of relevant pages of their report marked with notes. See DCE's phone log of talk with Richard Overton (14970,).

This is a response to CIRAD's request for comments on their report. It concerns only those sections in the report in which ARC is mentioned. 1

In general, there appear to us to be too many unsupported (and insupportable) extrapolations made from a small data base; we have not dealt wth most of them feeling that decisions to make such extrapolations are best left to the authors and critcisms of unfounded conclusions are best left to the future readers. 2

We also find ARC is often cited in conjunction with out-of-context, unattributed quotations possibly made by individuals who were not in a position to speak for the entire group and who were, in most instances, asked to take speculative positions on unfamiliar and inapplicable questions (cf., section 4.3.2.5, p. 86). These are used to support otherwise undefended recommendations and conclusions of the CIRAD authors. 3

Those sections in which we are mentioned and which are not included below are generally inoffensive to us though the reservations mentioned above sometimes apply. 4

These notes were prepared by Harvey Lehtman and Charles Irby. They were reviewed by Drs. Engelbart and Watson. 5

Section 3.2.5 Terminal Observations 6

Page 23 6a

Paragraph 2: "each of four programmers donated about two hours..." should read "each of four programmers donated about eight hours in two hour blocks..." 6a1

Paragraph 5:"The bug in the unmodularized version....to the calling program." should read: 6a2

The bug in the unmodularized version is the erroneous use of the word RETURN instead of the correct word EXIT in the phrase ON SIGNAL ELSE RETURN after the beginning of the first loop in the procedure process. The statement is also misplaced and should occur at the beginning of the inner loop. This bug causes a return to the calling procedure instead of the desired exit from the inner to the outer loop and continuation of processing in the outer loop. 6a2a

Page 27 6b

Arrow should go from marked statement down to the BEGIN after the inner loop. (SEE MARKED PROGRAM.) 6b1

Section 4.2.5.3 An Example of a Heirarchy 7

Page 62 7a

Paragraph 4: "177v" should read "177B (i.e., 177 octal)". 7a1

Paragraph 6: should read 7a2

The longer program follows a heirarchical structure itself; the rules for going from one level to another are programming conventions at ARC. While the ARC L10 compiler doesn't require the groupings found in the program, ARC programmers find it useful to have all statements in a compound statement block at the same level. Thus the particular form of loops and IF-statements found in the program satisfy the ARC standard. The ARC conventions are geared to the language used, an ALGOL-like block structured language developed at ARC using its compiler-compiler system Tree-Meta, and to the features of the NLS system which permit level truncation. 7a2a

(Your paragraph seems to contradict the quote on page 69; There are system programmer's conventions for grouping blocks by their level in an NLS file. These conventions were arrived at through careful study and development of the NLS system. The types of groupings used in a FORTRAN punch card deck are likely to be quite different. Moreover, it might be said that "research papers on understandable levels and structures, such as that in the first sub-section of "Perception and Groups"...", do not seem to take advantage of the work done at ARC in the use of heirarchical arrangement of files to facilitate understanding.) 7a3

Section 4.3.1 Some Procedural Factors 8

Some of the unattributed quotations are meaningless as written. 8a

Page 68 8b

Paragraph 2: "The processed" should be "The processes". 8b1

Paragraph 4: Last sentence should be: 8b2

Included in NLS are command algorithms for, among other things, file structure and text editing, dialog creation and cataloguing, printed output production, information retrieval, and the creation, compilation, and debugging of source code used to make new versions of the system. 8b2a

Paragraph 5: First sentence should be replaced by the following: 8b3

Files are organized in levels. Although the computer system (the NLS editor and file system and the compilers used in conjunction with them) does not impose any requirement for the use of a particular level in the heirarchy for any particulr piece of text, the "people" systems very often create such conventions. Thus programmers have decided, for example, that the source code which makes up a procedure should be one level down from the procedure name. Blocks and level choices in LOOP's, IF's, and CASE statements facilitate viewing (and reviewing when debugging) code both online and in the hard copy versions and follow suggested programming standards. The system allows a programmer... 8b3a

Page 69 8c

Paragraph 2: "... it makes you job much easier." should be "...it makes your job much easier." 8c1

Paragraph 3: "b. "You can figure out..."" makes no sense to us. Did someone really say that? Delete. 8c2

Paragraph 4: "The main value of one debugging tool..." also doesn't have a clear meaning. Delete. 8c3

Paragraph 9: "c. "Using BEGIN..."" should read 8c4

c. "Using BEGIN and END as the first and last statements of blocks in compound statement constructions." 8c4a

Paragraph 10: "It is handy..." makes no sense. Delete. 8c5

Section 4.3.1.4 Other Experimental Results 9

Page 74 9a

How "clear, objective," and reliable the findings in this section are is open to debate. 9b

Point 1-- Is the data base large enough to come to such a conclusion? 9b1

Secton 4.3.2.5 Digit/Symbol Spacing 10

Associating ARC with the statements and conclusions made in this section seems out of order. 10a

Page 86 10b

Paragraph 2: So what? Of what validity is it for someone here to "observe" that the findings "seemed reasonable" and be quoted? 10b1

Paragraph 3: "Of greated..." should be "Of greater..." 10b2

Paragraph 4: What? We can do machine searches for content. Just how would such "heirarchical search procedures" be used here? Why are there no examples here? The whole report is lacking in sufficient supporting material. 10b3

Section 4.3.2.7 Resonable Response Times 11

Page 88 11a

This section is completely unsubstantiated. Paragraph 2 is particularly objectionable to Doug. There is a continuum of responses one should expect in an interactive system. Thus, a particularly short time is desirable for a bug selection mark to appear on the screen while a longer time is expected for more complex tasks. The times quoted (1/10 and 1/4 second) are not results of published scientific research which we have done and which we would permit to be used in such a context. 11a1

Response to CIRAD Request for Comments on Their Report

(J15067) 13-MAR-73 15:28; Title: Author(s): Lehtman, Harvey G. /HGL;
Distribution: /dce rww chi dsk jdh ; Sub-Collections: SRI-ARC;
Clerk: HGL;
Origin: <LEHTMAN>RESPONSE.NLS;1, 13-MAR-73 15:18 HGL ;

JBN 13-MAR-73 7:40 15068

Reply to Account Number Query in 14999

Alex -- Account 3 is, was, and will be the account to use. The problem about account 1 was an error here. --Jeanne 1

Reply to Account Number Query in 14999

(J15068) 13-MAR-73 7:40; Title: Author(s): North, Jeanne B. /JBN; Distribution: /aam mdk ; Sub-Collections: SRI-ARC; Clerk: JBN;

T zero

trivality of trivalities
in "execute journal" "distribute document" after supplying a document number, one is ask "T0:" note that this is spelled upper case letter T zero. --jon. 1

I zero

(J15069) 13-MAR-73 10:39; Title: Author(s): Postel, Jonathan B. /JBP; Distribution: /JEW; Sub-Collections: NIC; Clerk: JBP;

jeannie,

relative to rfc#473 (nic 14811), we received a copy of a response from ken bowles at ucsd indicating they have both an interpreter for mix and the corresponding mixal assembler. dick watson was also copied on the letter if you would like to see the text. text on how to use should now be available on ucsd help program. they are host #35 and the logon procedure is zc1 password is arpaone. this may be an item for the arpanews update since it would introduce ucsd help capability to other network users. their help capability, in my opinion, is one of the nicer ones on the network. if you agree, let me know and i will call ken bowles to clear it. regards,jean 1

(J15070) 13-MAR-73 6:26; Author(s): Iseli, Jean /JI; Distribution: /JBN; Sub-Collections: NIC; Clerk: JI;

Another Try by Barden

Title: A Summary of What I Have learned about NLS at SRI 1

Preface This summary is based on the Command Summary, p. 63, et seq. of the TNLS Beginner's Guide (7 Aug. (1972). I shall ignore how to get in and out of the system and go directly to TNLS. 2

Chapter 1. This chapter and its sections are devoted, in general, to what a file is and how to work with it. The basic command is i[nsert] s[tatement at] ADDR Cr Cr. An early difficulty was finding the effect of the marker. It is now clear to me that in the case of a character, the insert skips a letter, then inserts; in the case of deletion of a character, it deletes what the lower > is pointing to. I had no trouble with the backspace controls, A, W, and Q. I have not used the control R, but do now. It didn't work. It is supposed to show the current LIT but I only had a ? mark for my efforts. 2a

Section 1. I really had no trouble with the null, load, and update file commands, nor with the Tenex file commands, such as directory and delete. 2a1

Section 2. The commands for showing where I am, such as /, , and ., gave me little trouble after a bit of practice. I did discover that one can combine these commands, and save some time. For instance a . with a statement number and a space, then a > command takes me directly to the end of the statement. And I learned how to add to the statement with the i[nsert] w[ord] command. 2a2

Section 3. Printing all or part of a file, except for viewspecs, was straightforward. I had to learn that the LF command is, on the display terminals, a control J. The arrow up and control O commands worked fine. O[utput] d[evice] t[eletype] commands gave no trouble, and I shall be using the similar command for the lineprinter. 2a3

Section 4. Editing by statement, involving the copy, move, replace, break, and append commands presented no problems, though I do not use them much. All these, plus the insert and substitute commands for characters, words, and text were, of course, important, and I am having plenty of practice with them. I did not actually use the substitute command across the entire file, but the manual example of TNLS for NLS makes this seem simple in the straightforward cases. Some experience with other editing systems makes me realize that substitutions across the file need to be carefully thought out. 2a4

Section 5. On the content editing, Marilyn Auerbach showed me how to turn on the / and commands with e[xecute] sh[ow] c[ontent] and e sh s[elections], which are fine for unsure beginners who do not yet have much confidence in their own commands. This became confusing as I went on, because I only want to see the / and printouts when I really am confused. 2a5

Chapter 2. The Journal commands e j, and s s, f, or m, presented no great difficulty, because things are programmed to prompt the user on messages. I think that ordinarily, I shall make an edited file, then commit it as a message. The only editing allowed in freehanding a message is the backspace. This is fine for a casual conversation, but it is a bit expensive for what most people have to say when freehanding their thoughts. I'm not quick that way and prefer a more finished statement. 2b

Chapter 3. Now there were several areas I learned about, most of them still mysterious. I shall simply mention that I have notes and such manuals and printouts as were furnished, with serious intentions of studying them. 2c

It is going to take me quite a while to learn about viewspecs especially in relation to the commands, substitute, execute assimilate, linking, and even printing. 2c1

I achieved a fairly good formal understanding of the tree structure invited by NLS, but learning to use it as a matter of manipulating branches, plexes and groups is going to take a while. 2c2

I think I did get a sound, elementary understanding of the Tenex file system as worked out here, mainly because I had the Case Tenex to compare it with. 2c3

The print directives like 14 MAR 73 Gpn;, and so on baffled me temporarily, but then all formatting has this effect at first encounter. 2c4

I had some fine help from Mrs. Jernigan and Marily A. on finding and handling the NIC files. I have some notes on the "goto" programs, expecially the content analyzer. I may eventually be able to do some of the things with preprogrammed links, which Marilyn A showed me. 2c5

Chapter 4. All in all, the instruction by Messrs. Kudlick and Van Nouhuys and Ms. Auerbach was first-rate and patient. I shall certainly be on the network practicing during such hours as you are on light loads out here--namely, 9-11 a.m. and 9-11 p.m. Cleveland time. Many, many thanks for the opportunity to start learning my way around. This, then was the message I tried to send that evening, with the wrong file on board. 3

Another Try by Barden

(J15071) 12-MAR-73 20:11; Title: Author(s): Barden, John P. /JPB;
Distribution: /DCE; Sub-Collections: NIC; Clerk: JPB;
Origin: <CASE-10>BAR.NLS;8, 12-MAR-73 20:08 JPB ;

LLL 13-MAR-73 12:56 15072

NAME CHANGE FOR CAROL HOFFMAN

For arc's information, Carol Hoffman has changed her name to Carol Guilbault (Gee-bow). 1

NAME CHANGE FOR CAROL HOFFMAN

(J15072) 13-MAR-73 12:56; Title: Author(s): Lane, Linda L. /LLL; Distribution: /SRI-ARC; Sub-Collections: SRI-ARC; Clerk: LLL;

Resource Notebook Stuff

Dave ... The meeting I had tentatively set for March 19th has
been postponed, with no new date until the selection
of "regional agents" --- as briefly described in the RFC
--- is completedd.
Mike Kudlick. 1

Resource Notebook Stuff

(J15073) 13-MAR-73 13:04; Title: Author(s): Kudlick, Michael D. /MDK; Distribution: /dhc ; Sub-Collections: SRI-ARC; Clerk: MDK;

On Site NLS Classes

Dirk ... Both JEW and HGL have seconded the idea of going out to the sites to teach NLS, and I think we should act now to begin to set things up. JEW's suggestion (see -- 14973,) that we send an RFC to solicit comments on the desirability of on-site teaching is a good one, it seems to me. Would you initiate this? Maybe (check with MFA) the Boston area is the right place to start (BBN). But there are new sites as well --- Dept of Commerce at Boulder, Range Measurement Lab in Florida (Mike Young's spot), ... Mike. 1

On Site NLS Classes

(J15074) 13-MAR-73 13:21; Title: Author(s): Kudlick, Michael D. /MDK; Distribution: /dvn mfa hgl jew rww ; Sub-Collections: SRI-ARC; Clerk: MDK;

MDK 13-MAR-73 13:52 15075

NIC Envelopes

Susan ... I like your idea about ordering pre-printed envelopes. Would you please prepare the necessary forms?

Before we actually spend the money, however, I'd like to see what it is we'll be buying. And I'd like you to estimate as best you can what amount of time it currently takes us to affix the NIC's rubber stamp. But since it will only cost us about $130 for the printed envelopes, please don't spend much time figuring out how much money we'll save; just make an estimate. Thanks ... Mike 1

NIC Envelopes

(J15075) 13-MAR-73 13:52; Title: Author(s): Kudlick, Michael D. /MDK; Distribution: /srl jcn rww dvn jbn ; Sub-Collections: SRI-ARC; Clerk: MDK;

Ongoing Journal Evolution: Review Status

To RWW, re "Ongoing Journal Evolution" 1

The items below were discussed briefly in a conversation between JDH and MDK on 13-MAR-73, on the general subject of current and near-future Journal needs and desires. 2

We agreed that categories 2) and 3), on "work to be reviewed" and "unresolved questions", would be the subject of one and possibly more design review sessions in the near future. 3

The first of these meetings will be held some time (to be announced) in the next two weeks. At this meeting, a schedule for further reviews would also be set. 3a

1) Work in Progress 4

Journal re-entry: 4a

to be able to temporarily leave the journal, do some work in NLS, and return to the journal. 4a1

options at delivery time: 4b

to allow the operator to specify idents and/or documents for delivery, with other documents not being delivered at that time. 4b1

↑C temporarily: 4c

to escape from the Journal to the exec temporarily, with return via "continue". 4c1

"recover files": 4d

to give high priority to the program that cleans up journal files after recovery from a crash, in order to get the job done and avoid being put on "queue-four". 4d1

load average graceful shutdown: 4e

to implement graceful shutdown of journal when load average gets too high; otherwise "queue-four" mechanism increases chances of system crash occurring while journal is in midstream. 4e1

2) Work to be reviewed 5

expanded "status" options: 5a

to be able to see and edit "message" contents. 5a1

acknowledgement when submission completed: 5b

to be told where the document is temporarily residing until journal delivery occurs, and to be told (via an automatic tickler file) that a given title has been accepted for delivery. 5b1

mods to citation format: 5c

to move the link to the first line of the citation (replacing the "number"), in order to make it possible to do " space up-arrow" in --- i.e., jump to link --- TNLS without getting caught on parenthetical expressions in the title. 5c1

3) Unresolved Questions 6

RFC distibution problems 6a

document number as part of distr list 6b

update initial files occasionally 6c

secondary distribution using TEJOURNAL 6d

multi-site journal: the impact of the utility 6e

privacy of files and catalog references 6f

Ongoing Journal Evolution: Review Status

(J15076) 13-MAR-73 17:34; Title: Author(s): Kudlick, Michael D. /MDK
; Distribution: /jdh chi dsk mdk rww jcn ; Sub-Collections:
SRI-ARC; Clerk: MDK;
Origin: <KUDLICK>JL.NLS;2, 13-MAR-73 17:31 MDK ;

Cngoing Journal Evolution: Review Status Additions

Errata in "Ongoing Journal Evolution: Review Status"
Should include in 3) these two addtl items ...
   the number system overriding titles, etc.
   the need to get TNLS and DNLS journal submissions compatible 1

Ongoing Journal Evolution: Review Status Additions

(J15077) 13-MAR-73 17:38; Title: Author(s): Kudlick, Michael D. /MDK; Distribution: /mdk ; Sub-Collections: SRI-ARC; Clerk: MDK;

To RWW, re CATALOG SYSTEM Plans. 1

The overall scheme for revisions to the Catalog System has two principal aspects: 2

1) No major new work will be done on the present Catalog System. 2a

2) A new Catalog System will be brought up in several stages. 2b

No New Major Work on the Present System 3

There are, conceptually, three distinct components to the Catalog System: 3a

a) data entry, verification, and preparation of a valid input file; 3a1

b) processing by the set of programs called "CPPPROGS"; 3a2

c) on-line and off-line viewing of the indexes prepared by these programs. 3a3

In my opinion, the processing that is done in step b) is the major bottleneck in the present catalog system. 3b

The reason for this is that the size of files and the type of processing are both unsuited for the PDP-10 Tenex. 3b1

Educated guesses place the cost of running the system at about 10 cpu hours per month (not including elapsed time, and not including delays for various overhead needs). This estimate may even be low. 3b2

The time spent in running is high, because of several factors: 3b2a

a) NLS files are time-consuming to process; 3b2a1

b) NLS does not process more than one file at a time, so that merging ("assimilating") the contents of two or more files can only be done manually in the present system; 3b2a2

c) The present system has two major aspects --- sorting and output processing --- which are particularly time-consuming on the PDP-10 Tenex/NLS system. 3b2a3

d) There are no fail-safe or restart provisions in the present system, so that a system crash after a couple of hours of running require almost complete re-running of the system. Such crashes are not uncommon. 3b2a4

The size of files is unsuited to the PDP-10 Tenex/NLS system because of two main factors: 3b3

a) Some files are too big for the system, and must be split into two files. This requires double processing in certain cases. 3b3a

b) The present scheme of directory space allocation is not flexible enough to handle temporary needs during production runs or re-runs. Not enough space is normally allocated to handle these temporary "overflow" needs. 3b3b

Several Stages to Reach a New System 4

Stage 1 4a

It seems clear to me (but I have no "proof" yet) that step "b)" processing can be done in a batch environment more cheaply than it is done here. 4a1

Therefore, a study on this aspect of the system will get the first priority, and will constitute "phase one" of the projected work leading to a new catalog system. 4a1a

Stage 2 4b

Assuming that the claim is borne out by the study, then a second phase, "phase two", will be comprised of the actual transition to a batch processing mode for step b) processing. 4b1

This will involve finding a suitable system on the Network (probably an IBM 360 or 370 configuration), and programming whatever code is required. The programming requirements are, generally speaking, four-fold: 4b2

Reading the NLS File 4b2a

This would be done either via a tape file created at the PDP-10, or by sending the file via FTP to the IBM 360 site. 4b2a1

Extracting and Sorting 4b2b

Sorting is well-parameterized and probably won't require extra coding. 4b2b1

Extracting the relevant information (like for NIC indexes vs ARC indexes) probably will require coding. 4b2b2

Formatting and Printing 4b2c

Formatting of printed output may be possible via a report program generator, though this has to be determined. 4b2c1

Preparing files for NLS 4b2d

Preparation of files readable by the PDP-10 Tenex system for inclusion back into the NLS file system (for on-line reading of the indexes) would also require some study and programming. 4b2d1

Stage 3 4c

Longer range, the efforts should be towards creating one integrated system that relies on the concept of resource sharing on a Network wide scale. 4c1

The resources to be shared are both processing power (as above), and storage facilities for large data bases. 4c2

Specific Objectives 5

Among the technical questions that require study under the above plan is the feasibility of accessing a reliable batch-oriented program suitable for catalog production. 5a

For this we need to determine: 5b

a) status of implementation of file transfer protocol; 5b1

b) status of remote job entry on IBM 360 or 370 installations currently on the Network; 5b2

c) criteria for selecting the most suitable site --- assuming both a) and b) get positive responses, i.e., assuming availability of FTP and RJE; 5b3

d) suitable sites within reasonable travel distance; 5b4

e) restrictions on network use (file size, processors, computer time) at those sites; 5b5

f) rates for using those systems. 5b6

With answers to these questions, we then will proceed to: 5c

a) select an implementation language; 5c1

b) schedule analysis tasks; 5c2

c) schedule development of a test program that transfers an NLS file from SRI to site X, compiles a short program at X, runs it against the file, and sends the resulting file back to us in NLS form. 5c3

(Note: b) and c) can proceed in parallel.) 5c3a

(J15078) 13-MAR-73 8:32; Title: Author(s): Vallee, Jacques F. , Kudlick, Michael D. /JFV MDK ; Distribution: /rww jfv wlb jbn jew mdk ; Sub-Collections: SRI-ARC; Clerk: MDK;
Origin: <KUDLICK>CAT.NLS;7, 13-MAR-73 8:30 MDK ;

Problems with ident submode

Why can I not, using Execute IDentification submode, and as coordinator of TUG (I am JDB not RST), make modifications to the ident record for TUG. I get the complaint "Must be cordinator" (sic). 1

RST 13-MAR-73 5:50 15079

Problems with ident submode

(J15079) 13-MAR-73 5:50; Title: Author(s): Tomlinson, Ray S. /RST; Distribution: /JBN; Sub-Collections: NIC; Clerk: RST;

TNLS Bug, Execute Journal, Distribute Document, Spelling Error in Prompt

Jon Postel (JBP) from UCLA-NMC reports (see -- 15069,) that the 'Distribute Document' subcommand of 'Execute Journal' in (I presume) TNLS prompts for the distribution list with 'T0:', where the second character is a zero, rather than the appropriate alphabetic. 1

JEW 14-MAR-73 11:53 15080

TNLS Bug, Execute Journal, Distribute Document, Spelling Error in Prompt

(J15080) 14-MAR-73 11:53; Title: Author(s): White, James E. (Jim) /JEW; Distribution: /BUGS; Sub-Collections: SRI-ARC BUGS; Clerk: JEW; Crigin: <WHITE>BUG.NLS;2, 14-MAR-73 11:49 JEW ;

Meeting Report -- NMDT meeting - March 12, 1973
Participants: CHI, JGM, CFD 1

Agenda: 2

(1) Discussion of feedback on conversion justification paper. 2a

The criticisms on the justification paper seem to extend from the fact that we did not directly address ourselves to the consideration of alternate conversion strategies. 2a1

Some of the alternative strategies might be:
a). Modify and extend L10 system to include more software engineering capabilities. 2a2

1) Extended control structures
2) Fully typed language
3) Defined scope of names. 2a2a

b). Make L10 and MPS linkages compatable, Allowing L10 and MPS to coexist. 2a3

After considering the potential impacts of either of the above mentioned alternatives, we conluded that neither had significant merits. 2a4

(2) Planning activities. 2b

The following list of activities was constructed and the interconnections were noted: 2b1

	Activity	Depends on	2b2
1).	Command language facility		
2).	file system		
3).	data base system	2,1	
4).	large character set	2	
5).	graphics	2	
6).	interface to external systems	1,2,5,8	
7).	programmable NLS	1	
8).	portrayal generator	2,3,4,5	2b3

Plans: 3

We decided to document our initial NLS model and concurrently propose a design documentation standard. This document should be available in draft form during the week of March 19. 3a

Next meeting: 4

Wednesday, March 14, 10:00am at SRI 4a

15081 Distribution

Irby, Charles H. , Dornbush, Charles F. , Mitchell, James G. , Paxton, William H. , Deutsch, L. Peter , Wallace, Donald C. (Smokey) , Satterthwaite, Ed H. , Bass, Walt , Andrews, Don I. , NMDT Meeting Report - March 12, 1973

(J15081) 14-MAR-73 8:39; Title: Author(s): Dornbush, Charles F. /CFD ; Distribution: /NMDT NMRT ; Sub-Collections: SRI-ARC NMDT NMRT; Clerk: CFD ;

Visit Log: 13 Mar 73, Howard Greyber, AAAS

Visit Log: 13 Mar 73, Howard Greyber, AAAS

Howard Greyber
American Association for the Advancement of Science
1515 Massachusetts Ave N.W.
Washington, D.C. 20005
(202) 467-4485 1

Howard is arranging a meeting for AAAS -- Feb 25 to Mar 2, 1974, in San Francisco. Mr. Anderson sent him through Bart Cox's area as one of the possible domins of activity within SRI that could provide content. Howard is also looking for people to manage the conference. Bart and Earle Jones had talked with him before I saw him (from 4:50 to 6:10). 2

He became fascinated in our substantive work, and we never did get to a dicussion of his conference plans. Gave him copies of: OSR1, FJCC68, dinosaur paper (5255,), IPT summary (13537,), and coordinated info servicess for communities (12445,). Also, printed out and gave him a copy of the NCC "Workshop" paper (14724,). 3

I assume that there will be follow-up action by Howard, probaby via other direct contact than with me; meanwhile ARC should consider its interest and position with respect to the AAAS conference. 4

Note: It occurs to me that the general scientific audience reached by AAAS might be an important one for ARC to consider communicating with, about such as knowledge workshops, central workshop services for discipline- or mission-oriented communities, etc. 5

Visit Log: 13 Mar 73, Howard Greyber, AAAS

(J15082) 14-MAR-73 9:07; Title: Author(s): Engelbart, Douglas C. /DCE ; Distribution: /rww jcn bc ; Sub-Collections: SRI-ARC; Clerk: DCE ;

Glad that Harvey pointed you to IMOL. I forgot about that. 1

Your understanding of the "←" construct is correct. 2

You will have to use lower case to write your assembler, since META requires rule names to be lower case identifiers (and it makes use of the difference). However, you will have no problems with your assembler accepting only uppercase. Just recognize all identifiers with .UID. 3

We pulled Knuth out and read about his n'here' labels. Our general response was not very favorable, but I think you will have little trouble inplementing it. 4

Try the following: 4a

If a label is undefined, there is no problem. References to it will be linked and presumably you will have a loader that can do fixups. A fixup will be produced when the label is defined. 4a1

If a label is defined and a reference to 'F' is made you have a problem. You want to produce an undefined link to a label (whose name is already defined). Do this by using a generated label. When the label is defined, define the generated label and redefine the symbol. 4a2

A 'B' reference to a defined symbol is trivial. 4a3

Let me know if my understanding of the n'here' stuff is incorrect and gets you in a jam. 4b

You can run the whole mess in hex if you wish by changing the number recognizing routine in the library. Thats the guy that's invoked by saying .NUM. I think the routine that actually does the evaluation is called numval. It's in <meta>libe.nls. 5

Getting hex numbers out is another problem. Meta uses NOUT. Does that work? 6

Your understanding of spaces is right. Any number of spaces, line feeds, tabs, cr's, or comments get smashed into one space in the input stream. 7

If you want to use % in yor assembler, you will have to change the library <meta>libe.nls. The comment recognition is done when blanks are deleted - done by delb. 7a

No one here can understand just why you don't write an MOL. You could write one that LOOKED like an assembler if the user didn't know about IF statements and BEGIN's and END's ( ) We have found MOL's very useful - a few very simple constructs make machine language MUCH easier to read. Incidentally, that our chief objection to Knuth's N'here labels- labels are evil because 90% of the time they are just obscuring simple LOOP's, but when they are all the SAME NAME HELP 8

Just found out that Harvey has answered you also. Well, anyway you're getting answered. 9

Don 10

(J15083) 14-MAR-73 11:36; Title: Author(s): Andrews, Don I. /DIA;
Distribution: /bpc ; Sub-Collections: SRI-ARC; Clerk: DIA;
Origin: <ANDREWS>COSELL.NLS;2, 14-MAR-73 11:19 DIA ;

Training Plan: Notice of DRAFT

My file (vanNouhuys,train,) is a draft of the long-pending training report and plan. It is a complete draft except for minor editing and filling in links which Kay should finish by tomorrow (3/15) 1

You are the training design team. 2

I ask you to consider whether this document will serve as a traing plan in response to (journal 14164,)and superceding (journal,14840,), or as the basis for such a training plan. 3

If either of you wants us to talk before going on, tell me and I will call a meeting. 4

If I do not hear from either of you by 3/20 I will forward the plan to the design review team and to certain other interested parties. 5

Namely: rww njn tfl and maybe some one at ARPA if JCN identifies an appropriate person. 5a

Training Plan: Notice of DRAFT

(J15084) 14-MAR-73 16:05; Title: Author(s): Van Nouhuys, Dirk H. /DVN; Distribution: /MDK JCN KFB(for your information); Sub-Collections: SRI-ARC; Clerk: DVN;

DVN 14-MAR-73 16:39 15085

What Kirk Kelley is Really Doing

The disparity of Kirk Kelley's title and pay with what he does is getting more and more acute. 1

He is participating in DEX desing (14912,) (14164,:;["KIRK"];k), is redesinging and maintaing locator (14861,), taking and active role in solving the implimentation problems of the SIGART Newsletter and its dialog facility (14716,)(14888,), is updating documentation, is teaching display and DEX (14860,), and providing other intellectual work (catalog,arcjaincnl,:;["Kelley"];i) clearly outsie the domain of clerk typist. 2

We should do something for him. 3

(J15085) 14-MAR-73 16:39; Title: Author(s): Van Nouhuys, Dirk H. /DVN; Distribution: /JCN RWW MDK KIRK(for your information); Sub-Collections: SRI-ARC; Clerk: DVN;

BPC 14-MAR-73 10:58 15086

More Tree-Meta Questions from Cosell@BBN-TENEX

Messages received from Bernie Cosell at BBN using sndmsg; entered into Journal.

More Tree-Meta Questions from Cosell@BBN-TENEX

14-MAR-73 4:36:20,2947 1

Net mail from site BBN-TENEX rcvd at 14-MAR-73 4:36:16 2

-------- 3

Date: 14-MAR-73 0103 4

From: COSELL at BBN-TENEX 5

Re: TREE META 6

cc: ANDREWS at SRI-ARC 7

- - - - 8

I'D LIKE TO THANK YOU BOTH FOR BEING SO RESPONSIVE AND HELPFUL. HELPFUL. 9

AS IT TURNS OUT, I'M FAIRLY FAMILIAR WITH BNF AND GRAMMARS AND ETC AND SO ONE OF MY MAJOR HURDLES IS TO CONQUER THE SYNTAX TO SEE HOW WHAT I KNOW HAS TO BE EXPRESSED SOMEHOW ACTUALLY IS (SPEAKING OF SYNTAX AND PARSING PROBLEMS). 10

PETER DEUTSCH'S PROPENSITY TO USE ALL CAPABILITIES OF ANYTHING HE 11

TOUCHES HAS MADE HIS <IMLAC>IMOL, TOGETHER WITH THE "SYNTAX AND 12

SEMANTICS" SECTION NEARLY ALL I NEED - I CAN FIND AN INSTANCE, 13

PRESUMABLY CORRECT, OF ALMOST EVERY CIRCUMSTANCE THAT I HAD 14

A QUESTION ABOUT. I RECOMMEND THAT YOU CONTINUE TO RECOMMEND THAT 15

FILE IF YOU HAVE TO FIELD FUTURE INQUIRIES. 16

I THINK I'LL PEEK AT THE SIMPLER EXAMPLE, JUST TO MAKE SURE I'M GETTING IT DOWN RIGHT. 17

A COMMENT - AS FAR AS I CAN TELL THE "*" CONSTRUCTION (I BELIEVE IT SHOULD BE IN SUBEXP =['←] #(STEST ['*] / NTEST) IS MISSING FROM THE SYNTAX DESCRIPTION. I'M ALSO FUZZY ON WHAT THE ← REALLY MEANS. AM I CORRECT IN GUESSING THAT IT IT USED WHEN AN STEST CANNOT BE UNIQUELY DEFINED SO THAT AFTER MATCHING THE FIRST ELEMENT HAS CLAIMED SOME OF THE INPUT AND THE REMAINDER FAILS IT CAN GO BACK AND TRY TO INCLUDE MORE OF THE INPUT IN THE FIRST ELEMENT ("FIRST" HERE IS BY EXAMPLE) TO TRY FOR THE DIFFERENT

MATCH? - MUCH AS IN A BOTTOM-UP PARSER ONE WOULD OCCAISONALY "BACKUP" TTO TRY A DIFFERENT REDUCTION ON AN INPUT CHARACTER. 18

WHAT KIND OF TROUBLE WILL I GET INTO BY ONLY HAVING UPPERCASE? 19

IN CASE YOU'RE INTERESTED - WHAT I'M UP TO IS TRYING TO BUILD A MORE OR LESS GARDEN-VARIETY ASSEMBLER FOR THE LOCKHEED SUE. I'D PLAY WITH BUILDING A MOL, BUT THAT'S FAR TOO RADICAL A CONCEPT FOR THE GROUP HERE. THE TWO MAJOR THINGS I'D LIKE THE ASSEMBLER TO HAVE IS A SYNTAX THAT REFLECTS THE "SOURCE-DESTINATION" STRUCTURE OF THE MACHINES INSTRUCTIONS (E.G. I'D LIKE "MOVE AC4->FOO(WDINDX)" TO BE THE RIGHT THING) AND TO SEE IF KNUTH'S N'HERE' TYPE OF LOCAL SYMBOLS ARE REALLY AS GOOD AS HE IMPLIES IN VOL I. ALSO, I'D LIKE THE WHOLE THING TO WORK IN HEX. 20

CAN YOU GIVE ME ANY QUICK HINTS ON HOW ONE MIGHT HANDLE THE 2H: JUMP 1F, TYPE OF SYNTAX? - 21

22

ALSO TWO OTHER LITTLE QUESTIONS - WHAT IS THE SYNTAX OF SPACES - AS FAR AS I CAN TELL THEY ARE IGNORED BEYOND ACTING AS SYNTACTIC SEPARATORS. ALSO IS % FORCED TO BE GOBBLED UP AS COMMENTS IN THE RESULTANT LANGUAGE - AS FAR AS I CAN TELL DEUTSCH SEEMS TO HAVE GOTTEN % COMMENTS FOR IMNLS WITHOUT ANYTHING IN IMOL TO HANDLE IT. ALSO, ANY WARNINGS OR HINTS AT TRYING TO BUILD A COMPILER TO DEAL MAINLY WITH HEX? 23

THANKS AGAIN FOR YOUR HELP, AND I'LL REALLY TRY TO NOT BOTHER YOU-ALL TOO MUCH. 24

BERNIE 25

P.S. SRI IS DOWN AT THE MOMENT - I WONDER IF THIS "QUEUED" AND DEFERRED MAILING REALLY WORKS. 26

BPC 27

------- 28

14-MAR-73 7:11:21,421 29

------- 30

Date: 14-MAR-73 711 31

From: BBN-NET 32

More Tree-Meta Questions from Cosell@BBN-TENEX

Re: NIC 14045 33

cc: ANDREWS 34

- - - - 35

I was looking at my copy of NIC 14045 (the TREE META Overview) and I believe that statement 1e5b3a (R1B) should read: ... $("+" exp :add[2] / ... - inserting the "exp". 36

bernie 37

P.S. - I am really still COSELL@BBN, unless I've made it already to being COSELL@SRI. My existence as BBN-NET is purely imaginary. 38

bpc 39

------- 40

15086 Distribution

Cosell, Bernie P. , Irby, Charles H. , Watson, Richard W. , Deutsch, L. Peter , Andrews, Don I. ,

More Tree-Meta Questions from Cosell@BBN-TENEX

(J15086) 14-MAR-73 10:58; Title: Author(s): Cosell, Bernie P. /BPC ; Distribution: /bpc chi rww lpd dia ; Sub-Collections: NIC; Clerk: HGL;
Crigin: <LEHTMAN>MESS.NLS;1, 14-MAR-73 9:23 HGL ;

Answer to Tree-Meta Questions of Bernie Cosell at BBN

Also sent via sndmsg.

Answer to Tree-Meta Questions of Bernie Cosell at BBN

Dear Bernie, 1

Deferred mailing really works. 2

Your request, coupled with some other requests we have had recently, has served as an impetus to get me to finish the program environment section of the report. It seems, however, that you are getting along quite well withoout it. 3

Some responses to some of your easier questions (responses to the more difficult ones will have to wait until Don or I can consider them): 4

> The "*" construction-- in this context is an ntest (non-testing element) which is described in detail. Essentially, it passes control to unparse rules which may put out code then collapses the tree branches on the processed. The "←" in this context does imply that backup will be invoked if any of the testing elements fail. See the detailed explanation. 4a
>
> By having only uppercase, you will find it difficult to write a program in the Tree-Meta language (reserved words, etc., depend on case.) Assuming, however, that you have written and loaded the program, your language specification for the source code to be compiled or assembled may be upper case only if you wish. (There are ways of getting both cases into NLS files from upper case only devices in NLS, though the use of a more flexible terminal is recommended (by me). 4b
>
> We all seem to feel an MOL is better than an assembly language and results in code which is just as efficient, but chacun a son gout. There should be no real problem in using Tree Meta for an assembler, however. 4c
>
> > There are several ways of going about dealing with the 2H: JUMP 1F, syntax. (generated labels, etc.) Don is preparing a more detailed answer on this point. 4c1
>
> As for spaces, they are ignored in the language specification if you want them to be; IDs are delimited by them (and other non-alphanumeric characters, for example. You may force your syntactic separator to be, for instance, at least 1 space by having 1$SP in the syntax rule, etc. (I just discovered this explanation is more complicated than necessary and that perhaps I am missing the point of the question.) 4d
>
> Comments are gobbled up when enclosed by % because of the

definitions of the library routines which recognize basic entities (ID, UID, etc.); These basic routines may be redefined if you wish. (Which reminds me to send you a listing of Tree-Meta in itself along with the library (in L10, the algorithmic language used here, the compiler for which was written in Tree-Meta-- incestuous, isn't it?) 4e

You are correct about the typo in the overview. 4f

Thanks, Harvey 4g

15087 Distribution

Cosell, Bernie P. , Watson, Richard W. , Irby, Charles H. , Deutsch, L. Peter , Andrews, Don I. ,

Answer to Tree-Meta Questions of Bernie Cosell at BBN

(J15087) 14-MAR-73 11:01; Title: Author(s): Lehtman, Harvey G. /HGL;
Distribution: /bpc rww chi lpd dia ; Sub-Collections: SRI-ARC;
Clerk: HGL;
Origin: <LEHTMAN>ANSWER.NLS;1, 14-MAR-73 9:43 HGL ;

Those Mighty Clouds of Joy Come Rolling In

It's my delight to tell you all that on Monday I became engaged. My fiancee's name is Vicki Peterson, who happens also to have been my sister's house-mate for the past eight months. Vicki's family is in Indiana, and it's there that we'll be married late this June. I just wish you all could share my joy in Christ on this occasion, and hope at least that if I seem to be a little out of touch for awhile, you'll understand why. 1

Those Mighty Clouds of Joy Come Rolling In

(J15088) 14-MAR-73 1:22; Title: Author(s): White, James E. (Jim) /JEW; Distribution: /SRI-ARC; Sub-Collections: SRI-ARC; Clerk: JEW; Origin: <WHITE>EA.NLS;2, 14-MAR-73 1:15 JEW ;

Response to 'Distribute Document' Bug Report

Jon-- Got your bug report (see -- 15069,) regarding the spelling error in the 'Distribute Document' prompt. 1

There is now a more-or-less formal mechanism for reporting NLS bugs, which consists simply of addressing a Journal article (like the one you sent me) to the ident 'BUGS'. 2

> It's a help if the title of the article is a consise and descriptive statement of the bug -- whether in TNLS or DNLS, which command, and nature of bug. 2a
>
> I've reported a number of bugs myself by this method during the past month and they actually seem to get dealt with, and a response is made to the reporter. 2b
>
> I invite you to try invoking this machinery in the future; I've done so already for this current bug (see -- 15080,). 2c

Response to 'Distribute Document' Bug Report

(J15089) 14-MAR-73 12:21; Title: Author(s): White, James E. (Jim) /JEW; Distribution: /jbp ; Sub-Collections: SRI-ARC; Clerk: JEW; Origin: <WHITE>JBPREP.NLS;2, 14-MAR-73 12:20 JEW ;

JEW 14-MAR-73 12:34 15090

What Happened to the Calico Documentation?

Abhai-- What ever happened to that documentation you were going to send out about Calico, etc.? Did you send it? Should I have gotten it by now? --Jim 1

What Happened to the Calico Documentation?

(J15090) 14-MAR-73 12:34; Title: Author(s): White, James E. (Jim) /JEW; Distribution: /akb ; Sub-Collections: SRI-ARC; Clerk: JEW; Origin: <WHITE>CALICO.NLS;2, 14-MAR-73 12:33 JEW ;

(arpanews-distribution) 1

server	addressee		copies	
bbn-tenex	(AAM)		18	
ucla-nmc			1	
Mitre	(EHF)			1
aberdeen	Ermalee r. McCanley		20	
usc	john melvin	1	3	
	john heafner	1		
	keith uncapher	1		
rml	mike young	1	2	
	ed schelonka	1		

1a

1b

(J15091) 14-MAR-73 17:26; Author(s): Iseli, Jean /JI; Distribution: /JBN; Sub-Collections: NIC; Clerk: JI;

JIM,

COULD YOU PLEASE CHANGE THE COORDINATOR FOR THE MITRE-TIP FROM PEGGY KARP, WHO IS NOW AT SU-AI, TO JI SO WE CAN MAINTAIN IT. ALSO, COULD YOU PLEASE ESTABLISH MRL AS THE COORDINATOR FOR THE CACI GROUP. THANKS FOR YOUR HELP; REGARDS, JEAN ISELI 1

(J15092) 14-MAR-73 6:56; Author(s): Iseli, Jean /JI; Distribution: /JEW; Sub-Collections: NIC; Clerk: JI;

Preliminary from Barden

Dear Marilyn Auerbach: Last evening I was at SRI, I had a file, BAR, which I wanted to send to DCE thanking you all for the seminar. In fact, I sent file JPB in which I had some of your stuff for printing out and taking along. The mail revealed this error. So a couple of days ago, I tried to forward BAR to DCE, and seem to have succeeded only with BAR's last statement. So much for explanation. I shall now try BAR on you, as you advised. Please express my sorrows to DCE for all the garbage. It would be nice if he never saw it. 1

Preliminary from Barden

(J15093) 14-MAR-73 8:19; Title: Author(s): Barden, John P. /JPB; Distribution: /; Sub-Collections: NIC; Clerk: JPB; Origin: <CASE-10>FMAR.;1, 14-MAR-73 6:17 JPB ;

Preliminary from Barden

Dear Marilyn Auerbach: Last evening I was at SRI, I had a file, BAR, which I wanted to send to DCE thanking you all for the seminar. In fact, I sent file JPB in which I had some of your stuff for printing out and taking along. The mail revealed this error. So a couple of days ago, I tried to forward BAR to DCE, and seem to have succeeded only with BAR's last statement. So much for explanation. I shall now try BAR on you, as you advised. Please express my sorrows to DCE for all the garbage. It would be nice if he never saw it. 1

Preliminary from Barden

(J15094) 14-MAR-73 8:28; Title: Author(s): Barden, John P. /JPB;
Distribution: /MFA MFA; Sub-Collections: NIC; Clerk: JPB;
Origin: <CASE-10>FMAR.;1, 14-MAR-73 6:17 JPB ;

Another Try

Title: A Summary of What I Have learned about NLS at SRI 1

Preface This summary is based on the Command Summary, p. 63, et seq. of the TNLS Beginner's Guide (7 Aug. (1972). I shall ignore how to get in and out of the system and go directly to TNLS. 2

Chapter 1. This chapter and its sections are devoted, in general, to what a file is and how to work with it. The basic command is i[nsert] s[tatement at] ADDR Cr Cr. An early difficulty was finding the effect of the marker. It is now clear to me that in the case of a character, the insert skips a letter, then inserts; in the case of deletion of a character, it deletes what the lower > is pointing to. I had no trouble with the backspace controls, A, W, and Q. I have not used the control R, but do now. It didn't work. It is supposed to show the current LIT but I only had a ? mark for my efforts. 2a

Section 1. I really had no trouble with the null, load, and update file commands, nor with the Tenex file commands, such as directory and delete. 2a1

Section 2. The commands for showing where I am, such as /, , and ., gave me little trouble after a bit of practice. I did discover that one can combine these commands, and save some time. For instance a . with a statement number and a space, then a > command takes me directly to the end of the statement. And I learned how to add to the statement with the i[nsert] w[ord] command. 2a2

Section 3. Printing all or part of a file, except for viewspecs, was straightforward. I had to learn that the LF command is, on the display terminals, a control J. The arrow up and control O commands worked fine. O[utput] d[evice] t[eletype] commands gave no trouble, and I shall be using the similar command for the lineprinter. 2a3

Section 4. Editing by statement, involving the copy, move, replace, break, and append commands presented no problems, though I do not use them much. All these, plus the insert and substitute commands for characters, words, and text were, of course, important, and I am having plenty of practice with them. I did not actually use the substitute command across the entire file, but the manual example of TNLS for NLS makes this seem simple in the straightforward cases. Some experience with other editing systems makes me realize that substitutions across the file need to be carefully thought out. 2a4

Section 5. On the content editing, Marilyn Auerbach showed me how to turn on the / and commands with e[xecute] sh[ow] c[ontent] and e sh s[elections], which are fine for unsure beginners who do not yet have much confidence in their own commands. This became confusing as I went on, because I only want to see the / and printouts when I really am confused. 2a5

Chapter 2. The Journal commands e j, and s s, f, or m, presented no great difficulty, because things are programmed to prompt the user on messages. I think that ordinarily, I shall make an edited file, then commit it as a message. The only editing allowed in freehanding a message is the backspace. This is fine for a casual conversation, but it is a bit expensive for what most people have to say when freehanding their thoughts. I'm not quick that way and prefer a more finished statement. 2b

Chapter 3. Now there were several areas I learned about, most of them still mysterious. I shall simply mention that I have notes and such manuals and printouts as were furnished, with serious intentions of studying them. 2c

It is going to take me quite a while to learn about viewspecs especially in relation to the commands, substitute, execute assimilate, linking, and even printing. 2c1

I achieved a fairly good formal understanding of the tree structure invited by NLS, but learning to use it as a matter of manipulating branches, plexes and groups is going to take a while. 2c2

I think I did get a sound, elementary understanding of the Tenex file system as worked out here, mainly because I had the Case Tenex to compare it with. 2c3

The print directives like 15 MAR 73 Gpn;, and so on baffled me temporarily, but then all formatting has this effect at first encounter. 2c4

I had some fine help from Mrs. Jernigan and Marily A. on finding and handling the NIC files. I have some notes on the "goto" programs, expecially the content analyzer. I may eventually be able to do some of the things with preprogrammed links, which Marilyn A showed me. 2c5

Chapter 4. All in all, the instruction by Messrs. Kudlick and Van Nouhuys and Ms. Auerbach was first-rate and patient. I shall certainly be on the network practicing during such hours as you are on light loads out here--namely, 9-11 a.m. and 9-11 p.m. Cleveland time. Many, many thanks for the opportunity to start learning my way around. This, then, was the message I tried to send that evening, with the wrong file on board. 3

Another Try

(J15095) 14-MAR-73 8:36; Title: Author(s): Barden, John P. /JPB;
Distribution: /MFA; Sub-Collections: NIC; Clerk: JPB;
Origin: <CASE-10>BAR.NLS;8, 12-MAR-73 20:08 JPB ;

1

(J15096) 14-MAR-73 8:40; Title: Author(s): Barden, John P. /JPB;
Sub-Collections: NIC; Clerk: JPB;

The Results

I looked and believe I am still getting through only with the last statement in the BAR file. 1

(J15097) 14-MAR-73 8:42; Title: Author(s): Barden, John P. /JPB; Distribution: /MFA; Sub-Collections: NIC; Clerk: JPB;

What are your feelings on celebrating birthdays at ARC?

I would like some feedback on whether most of us enjoy getting together and sharing refreshments on birthdays and whether you would like to help provide refreshments.

Your suggestions are not only welcome, but also are needed.

One suggeston is that each person take care of the following birthday.

Please let me know how you feel. 1

(J15098) 14-MAR-73 16:17; Author(s): Lane, Linda L. /LLL;
Sub-Collections: SRI-ARC; Clerk: LLL;

NIC Xeroxing Workload

Susan ... Thanks for your very useful study on our xeroxing workload and costs (14979,). I think the key phrase you wrote is, "as long as we continue distributing in the manner we do now." Before we order any more equipment I would like to discuss and explore the possibility of not distributing anything except something like a daily "initial file", i.e., distributing citations to documents but not the documents themselves. There would be some exceptions, of course, such as mail to the PI group, or other "urgent" mail. But I think it would be possible to have people request their mail in hardcopy depending on their interests. This would reduce our xeroxing workload, if it worked, and that's why I'd like to look at it as a possible alternative before we order more equipment. If you have ideas along these lines, please let me know. 1

NIC Xeroxing Workload

(J15099) 14-MAR-73 13:09; Title: Author(s): Kudlick, Michael D. /MDK ; Distribution: /SRL DVN JBN RWW JCN LLL KIRK ; Sub-Collections: SRI-ARC; Clerk: MDK;

NJN 14-MAR-73 6:28 15100

response to tip gripe 13-mar-73 1648

A TIMEOUT from the TIP implies that the host (SRI in your case) has its ready line up indicating that it is alive, but its NCP or its logger is rejecting--maybe all its network ports are filled up. A DEAD REFUSED message can mean either the host is down (i.e. took its ready line down) or that the imp is down or unreachable. If SRI was struggling on its last legs you might get a TIMEOUT and then a DEAD REFUSED when it subsequently went down. You would get only DEAD REFUSED messages after that however, until the host came up again.

An alternating sequence of TIMEOUTs and DEAD REFUSEDs would probably mean a problem in the network with lines going up and down. Things get backed up then and it is either difficult to get your RFC across the country (in which case you will get a TIMEOUT) or impossible because of a network partition (in which case you will get a DEAD REFUSED).

One more possibility is that the host is having difficulty coming up after a crash, and could be bringing its ready line up and down even though its logger is not yet ready to operate. Those problems usually clear up after about 15 minutes (sometimes less and sometimes more.)

If you want to pursue this further you can send a tip gripe next time it happens noting the sites involved, the time, and the sequence of events. I can then report back to you what happened then. 1

response to tip gripe 13-mar-73 1648

(J15100) 14-MAR-73 6:28; Title: Author(s): Neigus, Nancy J. /NJN; Distribution: /IWC; Sub-Collections: NIC; Clerk: NJN;

Walt Bass 15 March 1973 15101
DIALOG-SUPPORTED DEBUGGING SYSTEM

Here's a think-piece on a possible addition to ARC's acronym soup -- DSDS. These ideas are still in pretty rough form, and your feedback will be appreciated. The printed document is about 20 pages long, so please give the system a break by picking up Xerox copies from my office. -- Walt

INTRODUCTION

This document is an initial think-piece on new techniques for system debugging which could be implemented in future versions of NLS. 1a

I think it represents one step towards a natural and perhaps necessary integration of DSS and SEAS concepts into a new discipline for collaborative system development, maintenance, and evolution. 1b

For purposes of identification I have tentatively lumped these techniques together under the name "Dialog-Supported Debugging System" or DSDS. 1c

This document does not pretend to be a design for a DSDS but does provide the beginnings of a framework for DSDS evolution. Your criticisims, refinements, and extensions of the ideas presented herein are cordially invited. 1d

MOTIVATION

We are at the threshold of a really new era in information system development. If our current thrusts in the areas of networking, collaborative system development, and formation of communities oriented towards collaborative program and data-base development bear fruit, we will find ourselves living in an environment in which the average user will be accessing, in the course of a normal day's work, programs and data bases which have been created by tens or hundreds, and eventually thousands or millions, of individuals. 2a

In this environment our present ad-hoc method of handling "bugs" and "errors" will quite simply break down -- users can not possibly know where to direct the "blame" for an error in the attempt to get it corrected, and there will be no one person (or group) which can be expected to know enough to find and fix bugs occurring anywhere within a world-wide system of computational and informational resources. 2a1

(Of course, we would expect to see the emergence of highly-specialized trouble-shooters who would be called in, with their own high-powered tool kits, to fix the most untractable and costly bugs -- but in this document we want to outline methods by which the majority of conditions could be handled in an augmented way at much lower cost.) 2a1a

Even in our present state of relative isolation, we can see the seeds of these problems developing: when a system like the Catalog Production Processor or the Journal Delivery System blows up, it takes a knowledgeable and experienced system programmer to track the error down even to the point where he can identify the subsystem in which it occurred -- so that he at least can guess who to see for help if he can't fix it himself. 2a2

Moreover, the amount and complexity of "state information" needed to replicate an error condition are frequently so great that it becomes very expensive, both in programmer and system time, to recreate the condition so that it can be analysed in a debugging mode. This would indicate that there is high payoff to be gained by providing system tools which make it possible to automatically analyse and record enough of the state information to (1) identify the most likely source of the problem and (2) communicate this information to some person who is competent to fix bugs occurring in that area. 2a2a

Before going on, I'd like to clarify what problems I am addressing in this document, because I am not concerned exclusively with "bugs" which are the result of coding a program incorrectly but more importantly, perhaps, with "bugs" which occur when a user attempts to do something slightly different than was envisioned when a program or data base was created -- i.e., with new "cases" which are reasonably within the scope of the original design, but which are not handled correctly. 2b

Using this expanded concept of "bug," the DSDS becomes not merely a tool for fixing incorrect programs and data bases but a key part of a new discipline of system development in which systems are viewed as undergoing continuous evolution in response to the needs of a user community. 2b1

We know from painful experience that it is close to impossible to write programs which will do everything for everybody, and do it perfectly, as soon as they are released to the user community, and that frequently it is impossible to predict at implementation time just what needs should receive highest attention -- just which cases should be handled first, best, or at all. 2b1a

It also seems clear that we are enterring an era in which, as the sizes of user communities grow into the hundreds and thousands, the total investment in user training and methodology knowledgability will become comparable to the investment in system programming. 2b1b

This will mean that it will become increasingly unthinkable to conceptualize system development as occurring in distinct "stages" in which entire user systems are obsoleted and replaced overnight -- rather, we will be living in an environment in which users will expect all their needs to be satisfied within the existing "system" in a smooth, evolutionary progression. 2b1b1

Thus, we see that the concept of "bug" fades into the concept of "need" in the sense that what may appear to the user to be a bug -- the system didn't do what he wanted, expected, needed it to do in a certain case -- may appear to the programmer to be a need for an expanded system feature, and "DSDS" can be seen to mean "Dialog-Supported Development System" as well as "Dialog-Supported Debugging System." 2b1b2

It is important to emphasize at this point that everything which is said about programs in the context of system development and debugging applies equally to data bases. 2b2

In fact, as time goes on, I expect that data base development and programming will become increasingly recognized as complementary and co-equal partners in the process of developing very large information systems. 2b2a

At ARC we have witnessed over the past two or three years a growing commitment of resources to the development and maintenance of data bases and of tools for interacting with these data bases -- e.g., the Catalog, Identification System, Network Resource Notebook and Query, and various Documentation projects which could someday become part of an integrated Handbook System. An error or omission in any of these data bases can be fully as damaging or inconveniencing as a bug in one of our operational systems, and in many cases (e.g., Identfile, Journal Catalog Number System, etc.) the data bases are an indispensable part of our operational systems themselves. 2b2b

I certainly expect this trend to continue, particularly as our user community grows to include increasing numbers of non-programmers who are more concerned with -- and qualified for -- referencing and refining various data bases than they are with upgrading systems programs. 2b2c

In fact, one way in which programmers will best be able to serve these applications-oriented users will be by making systems which can be expanded in functional power and completeness through data-base manipulation as well as through overt programming. 2b2c1

Or, looking from a slightly different viewpoint, programmmers will be able to augment their own ability to implement large, increasingly intelligent, software systems by creating core programs which can be extended in their power and applicability by non-programming oriented users through data-base building.

In any event, the point I wish to make here is that precisely the same tools and methodologies that are needed for collaborative program systems development will be useful in the collaborative development of large data bases, and the design of a DSDS or any other component of a SEAS should take the needs of these complementary processes into account. 2b2d

CHARACTERITICS

Now that we have portrayed the environment in which a DSDS would be expected to operate, let's examine some of the ways in which such a system might be used. 3a

In this context, we'll be discussing modes of operation which range from ones that could be implemented in NLS as a methodological use of existing features through ones that involve a disciplined approach to new systems design and development to ones that require the developement of intelligent tools for automatic error analysis, reporting, and correction. 3b

What all of these modes share in common is their orientation to a system environment characterized by very large, collaboratively-developed programs and data bases, where the problems of coordinating the interaction of the community of system developers are as complex as the technology itself. Although any of the techniques described in the following sections would be of value even to an individual user working on his own programs and data bases over a period of time, the real cost/benefits payoffs are going to come only in the more socially intricate situations. 3c

DEBUGGING BY PROGRAMMER-INITIATED DIALOG

Let's start with an type of debugging that makes sense even in the context of our existing NLS. 4a

When a programmer writes or modifies an important system program, he must make choices on the basis of assumptions about what "cases" of some situation are most common or even likely to occur at all. These choices then determine which cases will be handled most effectively and efficiently and which cases will be considered to be "errors" if they occur. 4b

Often there is no way of making these choices other than by educated guess, and the programmer might wish that he could analyse the usage patterns to determine what the optimum choices might be before writing a program -- but the program's usage can not usually be analysed until it is written and becomes part of a user environment in which such analysis can take place 4c

Since this analysis can't take place a priori, there should be easy to use tools available for performing the analysis in actual operation of the system and for reporting the results of the analysis to the programmer when enough information has been collected to be of use to him. 4d

What I envision here is a straightforward modification of the Journal system that would allow messages to be sent by programs as well as by users. 4e

> For example, if a programmer decides that some case is extremely unlikely to occur -- like a file name more that 45 characters long, or a file with more than 12 levels, or two users simultaneously accessing the same file, etc. -- he might elect to defer writing code to handle that case until there is evidence that it actually has occurred (not to explicitly endorse this kind of coding practice, but to recognize that it does exist and will continue to exist as long as there is more programming work to be done than there are programming resources to apply to it) 4e1

Then, in addition to writing out an appropriate error message to the offended user, the program would also initiate the sending of a message to its author (or to some ident that would guarantee delivery to a person responsible for maintenance of the code involved). This message would be designed to contain enough information to identify the nature of the situation and the programs involved and might even contain the ident of the user who "discovered the bug" so that he could be sent a note acknowledging that someone knew about his problem and advising him that it would be fixed by such a time (or that it would not be changed and that he would have to do such and so to get around it). 4e2

Besides working in this kind of crisis mode, this feature could be used in places where the programmer wishes to collect some kind of usage statistics and then send a "report" to himself when enough information has been collected to enable him to complete (or redo) optimization of a certain piece of code. 4e3

Note that this particular kind of dialog-supported debugging does not require any artificially-intelligent programs or even an advanced overall system-design methodology and is merely an extension and application of existing dialog support system technology to a new domain. 4f

This has both the advantage of being a tool which we (ARC) could easily implement and experiment with and the disadvantage of providing debugging help only when explicitly programmed for -- or "you get what you pay for." 4f1

It is, however, a possible first step in the direction of a new system-design methodology in which the system itself is expected to be increasingly responsible for monitoring its own operation and automatically communicating needs for attention to an appropriate agent. 4f2

I say "agent" because the report may meed to be directed not simply to a given individual -- who could quit, die, or be relieved of responsibility for a certain program -- but more generally to an entity capable of carrying out a particular function or role. This entity could be a person, an organization, a functional role that (always) is assigned to a specific person or organization, or (in the long run) even another program or system. 4f2a

DEBUGGING BY USER-INITIATED DIALOG

In the previous section we discussed a DSDS technique which would enable a programmer to set up his programs to report to him when certain pre-specified conditions occurred. Another useful source of debugging dialog can be expected to originate from a system's user community -- a potentially large and far-flung group of people and machines. 5a

In this context we all recognize most users' complete willingness to complain loudly about system problems which bug them as well as their almost complete unwillingness to follow any procedure for bug-reporting which requires a disruption of their immediate work flow -- and most bugs not reported "on the spot" are forgotten until they occur again, resulting in a wasteful loss of potentially valuable information. 5b

This, coupled with the difficulty a user has in tracking down a sympathetic and knowledgeable ear, not to mention the programmers' own difficulties in keeping track of the status of the myriad of bugs which may be known, reported, or even fixed-in-the-next-version at any time, make system users a far less valuable source of feedback than they could be. 5b1

So, let's consider how the user community can be integrated into a DSDS in such a way that, not only does their feedback become a valuable part of the system development process, but also the frictions which inevitably develop between user and server communities can be systematically reduced. 5b2

One of the prime user-features requirements of the system we are seeking to characterize is that it make it easy for a user to give needed feedback about the system when it is most effective for him to do so -- i.e., when a "bug" has just occurred, and the circumstances surrounding the occurrence are fresh in his mind. 5c

This is predicated on the principle that it is important to the system developers to get feedback from the users and that, consequently, the system should be designed so that users will "feel good" about reporting bugs -- that it shouldn't cost them much in time and inconvenience to do so. 5c1

This implies that the bug-reporting mechanism should be invoked as an "escape" from normal system operating modes -- i.e. that the user can go into the bug-reporting mode in such a way that his state information is preserved intact. There are two significant reasons for this: 5c2

(1) The user can "resume" from the bug-reporting mode without having to pay heavy "set-up" costs to get back to the point where he left his work to report the bug. 5c2a

(2) The state information is available and can be analysed by the bug-reporting mechanism to derive data to be communicated in the bug report -- such as the names of files, sub-systems, non-standard modules, dialog linkages, network connections, system modes, etc. being used when the bug occurred. 5c2b

When a user invokes the bug-reporting mechanism, he should be confronted with a system which is evolving to be increasingly intelligent about interacting with him to properly characterize the error condition which is being reported. 5d

The system should have enough built-in knowledge about the kinds of bugs which can occur to be able to accept a terse statement of the problem when that is sufficient to convey the necessary information and to prod the user for additional information when he needs help in adequately describing what is wrong. 5d1

There are three items of information which this system must put together to be successful in its mission: 5d2

(1) The identification of the user (including his location, terminal connections, and the time of occurrence of the error) 5d2a

(2) The nature of the error which is being reported 5d2b

(3) The identification of the "debugger" that must be notified of this particular error 5d2c

The user's identification is needed to complete the characterization of the context of the error condition being reported and to make it possible to feed information back to him regarding what action has been taken in response to his report. 5d3

Even a primitive DSDS should be able to obtain this information automatically without having to ask the user for it, just as our current Journal system deduces the (default) author from the TENEX "job" state information. 5d3a

Items (2) and (3) interact because the system must determine who to report the bug to on the basis of its characteristics and circumstances. 5d4

This could be done after the user disconnects from the bug-reproting system, but only if enough information has been extricated from him to permit a determination on the basis of the collected information alone -- and in many cases the system probably will not be able to tell if it has enough information without actually making this determination. This is an area which will require study, because, if it takes too much time to make this determination, users will get so impatient with the system that they will stop reporting bugs. 5d4a

In cases where the user's characterization of the bug has inherent in it a precise description of the location of the bug within the system -- e.g., "The Resource Notebook says that MIT-AI has a PDP10, but they don't" -- it will be relatively easy for the system to determine who the bug report should be sent to. However, in the general case, this determination will require a non-trivial computation; this problem will be treated in the section on "Debugging by System-Initiated Dialog". 5d4b

In addition to interacting with users, the bug-reporting system must interact with system programmers and data-base managers so as to guarantee that bug reports reach their proper destinations and receive the necessary attention. Some of the ways an advanced DSDS might provide augmentation in this area are described below. 5e

If the debugger sucessfully corrects the bug, he should be able easily to "sign-off" the bug report to have it removed from his list of waiting needs and to have an appropriate message transmitted to the bug reporter. 5e1

If a bug report reaches the "wrong" destination or if, after analysis, the debugger concludes that the problem lies in a different area than was initially diagnosed, he should be able to interact with the bug-reporting system to provide any additional information which he may have to contribute and to help determine who is the most likely candidate for subsequent action. 5e2

Ultimately, in a system in which you might expect many users to discover and report the same bugs almost simultaneously, the DSDS should be able to determine if a bug has been previously reported so that it won't need to send duplicate reports to the debuggers. 5e3

Depending mainly on execution-time consideraions, the determination that a report is a duplicate could take place either at time of submission -- in which case the reporter could be informed that the bug has already been reported (although there doesn't seem to be any good reason to spoil his sense of "civic duty" in this way) -- or the decision could be made at delivery time. 5e3a

In either case the system would add the new reporter's ident to the list of those to be notified about action on the bug without creating any new work for the debugger. 5e3a1

In some cases a bug will have been not only reported but also acted upon (but not fixed) by a debugger. In these cases a message could automatically be sent to the reporter informing him of what the status of that bug is -- fixed in experimental but not running versions of affected system, acknowledged as a bug but can't be fixed, acknowledged and scheduled to be fixed, or whatever. 5e3b

This mechanism will undoubtedly also serve as a "help" mechanism, giving useful information about common user errors which often are incorrectly -- or impatiently -- reported as bugs. 5e3b1

In this case the system might keep track of the number of times this situation arises; then, if the number exceeds a pre-specified limit (or rate), a message could be sent to the system designers reporting that a certain user error is common enough that consideration should be given to redisigning the related user-features so as to make them more fool-proof.

There will, of course, have to be ways for the user to override these automatic mechanisms if he beleives that the system has misunderstood what he was trying to report. 5e3b2

One possibility, which requires an absolute minimum of new technology, would be to have a person that would be available "all the time" with both on-line terminal and normal telephone communications, who could be called or linked-to by a user who was having difficulty.

This might be the same person-role whose availability is advertised for giving assistance to users in using the system or it could be a seperate specialist who would be available at a moment's notice to take down a user's bug report.

This person should be knowledgeable enough about the systems involved to be able to prompt the user to give sufficient information to characterize the difficulty adequately to allow for follow-up.

What we have been describing in this section is one example of a broad class of subject-matter-specific dialog-support systems which we should expect to see evolve with the development of large collaborative on-line communities. 5f

What characterizes this class of systems is the existence of a "heated" dialog within a well-defined subject area among a large number of users which tends to "focus" (either always or intermittently) on certain participants in the dialog who can not be expected to reply individually to all communications that are directed at them. 5f1

In this kind of dialog, at least the "principals" at the focus of the dialog need to have augmentation in sorting out essentially duplicate communications for automatic handling so that they can concentrate on replying to "new" inputs. 5f1a

An elementary example which should help to clarify what I'm talking about would be a survey-taking dialog (such as a Delphi survey) in which the participants, or certain designated principal investigators, would be directing questions to the community that would result mostly in very stylized replies -- which could be automatically tabulated and/or responded to -- but there would be a few "unique" communications coming in, requesting clarification or providing information not available to most participants, which would have to be sorted out of possibly thousands or millions of less "interesting" replies. 5f1b

It should be clear that DSDS development would not only benefit from general DSS developments but could turn out to be a major driving force for extending DSS capabilities into new functional areas. 5f2

DEBUGGING BY SYSTEM-INITIATED DIALOG

In previous sections we have described ways in which dialog explicitly initiated by system programmers and users could be integrated into the debugging process. In this section we will attempt to determine the conditions under which debugging dialog can be initiated automatically by an operating system. 6a

We should first be clear about how system-initiated dialog differs from programmer-initiated dialog, for they are very similar in many respects. In fact there will undoubtedly be cases in which either term could be applied with equal righteousness, for we are not trying to define two seperate systems but rather to describe two methodologies for using the capabilities of a single integrated system, and it is clear that the system can also be used in other ways incorporating elements of both methodologies. 6a1

Programmer-initiated dialog can be characterized as specific, algorithmic, and expected -- a programmer provides a specific message to be sent to a specific addressee (most likely himself) under specific (algoritmically determined) circumstances, and he lives (for a while at least) under the expectation that this message may in fact be delivered. 6a2

System-initiated dialog, on the other hand, is more general, heuristic, and unexpected -- although it is provided for by programmers, it is designed to be invoked under a more general set of circumstances, to provide messages whose contents and addressees are more heuristically determined, and to create dialog which, although "expected" in its general nature, is likely to be "unexpected" in its specifics. 6a3

In other words, with programmer-initiated dialog the initiation conditions, the message, and the addressee are all pre-specified in detail, and all that the DSDS has to worry about is delivering the message when requested; while with system-initiated dialog the DSDS must also help determine when dialog is needed, what needs to be said, and who it should be said to. These three questions, WHEN, WHAT, WHO, characterize the areas of capability development required to provide system-initiated dialog within a DSDS. 6a4

First let us consider the question of WHEN a system should be expected to initiate debugging dialog automatically. This question is not as trivial as it seems, and I suspect that quit a bit of "fine-tuning" will be required to produce a system in which dialog is both initiated in most cases where it is needed and also NOT initiated in most cases when it is unnecessary. 6b

In general we could say that a system should initiate dialog whenever it becomes aware of a condition that should not exist and that the agents responsible for the system are not aware of (so far as the system knows). We have discussed the problem of handling multiple reports of the same bug in previous sections, so let's concentrate here on how a system becomes aware of error conditions. 6b1

In current programming systems this awareness is generally distributed throughout the entire system code, manifesting as the results of various test instructions and machine-interrupt operations. There usually is no particular piece of code which you can point to with the knowledge that all the system's awareness resides there -- i.e., there is no specific program into which you can patch a dialog-initiation process that will handle all conceivable error conditions. 6b2

Therefore, to make system-initiated dialog possible, all these points of error awareness must be tied together in some way. This could be through some kind of SIGNAL system which allows errors to be handled in a uniform way, or it could be expressly through the DSDS, with individual dialog-initiation processes at each awareness point as in programmer-initiated dialog. 6b3

Whatever mechanism is used, there is a clear implication that system-initiated dialog should not be considered an "add-on" feature but rather should be designed into a system from the start (or at least should influence the design of error-handling procedures so as to be implementable later on). 6b4

In particular it is important that all the information that is available at the time the system becomes aware of the error condition remain available also to the dialog-initiation process, for the awareness that an error condition exists does not necessarily embody in itself an awareness of the effective cause of that error. 6b5

Once the DSDS becomes aware that an error condition exists, i.e., when it is invoked by a part of the system which is aware of the error condition and has decided that a report should be made, it must decide WHO to report the error to and WHAT information to convey in the report. 6c

Determining WHO to report to requires two steps: 6c1

Locating the "effective cause" of the error condition. 6c1a

Finding WHO is responsible for maintaining the code or data base which is fingered as the effective cause. 6c1b

"Effective cause" is a subtle concept which will require much refinement as the DSDS evolves. Perhaps the best way to approach a definition is by analogy. 6c2

In the course of living in a civilized society people are in a continuous process of action. Inevitably some person will carry out a course of action which results in a situation that is recognized as "wrong" by someone else; then, it becomes a matter of law to analyse what evidence exists of the offender's life action in the attempt to isolate a specific act which violated the law along with the specific law which was violated. 6c2a

If we take the liberty of associating programs with people and error condition with wrongful situation in this little analogy, we can see that the error condition itself is not the thing which it would be most useful for the DSDS to report but rather the system "law" which was violated and the circumstances ("act") of the violation. Even after an error condition has been identified, much "detective work" may be required to track down the effective cause of the error -- the place where it is possible to fix the bug to prevent the error from recurring.. 6c2b

In a programming environment the "laws" themselves are usually fluid enough that a DSDS should probably treat both the law and the violation as the "effective cause" for purposes of reporting -- i.e., if an error condition results from some datum overflowing an allocated store, both the person responsible for allocating the store and the person responsible for creating the datum should be informed of the error. 6c2c

With current programming practice, the "laws" by which systems are supposed to operate are explicit only in the heads of the designers and occasionally in some documentation but merely implicit in the programs making up a system, and it seems that making a DSDS smart enough to track down effective causes with any accuracy could be a rather difficult task. 6c2d

However, if we project forward a few years to the day when Automatic Programming techniques have been developed to the point that they are useful for the collaborative design of large systems, we may find that all the information required for identifying effective causes has been quite explicitly linked into the actual operating programs and that relatively simple algorithms can be used in most cases. 6c2d1

In fact it should become increasingly possible to automatically determine the relative "solidity" of the law and the violation which have been isolated as effective cause of an error -- i.e., if the law is one of the basic properties of the system, probably only the violator needs to be notified; however, if the law is easily changed, the DSDS may reccommend that the law be changed to make the violation legal. Someday we'll even reach the point where many bugs can be fixed automatically without invocation of the dialog mechanisms at all, or with the dialog mechanisms used merely to record and report that something has been changed.

It is in this future Automatic Programming world that I would expect system-initiated dialog to come to full fruition, and one of the basic purposes of this document is to bring the concept of Dialog-Supported Debugging into awareness soon enough to influence the course of evolution of the Automatic Programming idea. 6c2d2

Once the effective cause has been located, determining WHO to notify should be a reasonably straightforward process. 6c3

Part of the discipline for the collaborative developement of large systems must lie in the area of associating with all code and data integrated into a system information about who is responsible for that code or data. 6c3a

We have the precursors of such a system in the NLS statement signature mechanism. However, statement signatures are only part of the picture, for there must be more explicit ways of recording who is responsible for actual maintenance of code and data as well as who was responsible for their creation. 6c3a1

We will problably witness the evolution of general tools, along the lines of our current identification system, for defining structures of functional roles -- both in the area of system debugging and elsewhere -- which would be assigned at any given time to either a specific entity or to another functional role (with the chain of assignments eventually ending at a specific actual entity). 6c3a2

These responsible entities could be either people or organizations (in which case the organization would itself be a structure of functional roles leading to real people) or even mechanical systems which were intelligent enough to carry out all or some significant part of the indicated function. 6c3a3

With such a tool for keeping track of functonal roles and relationships, it would be possible to record with all programs and data bases functional idents which would retain their validity even if a given real person died or vanished from the scene. 6c3a4

Even if such a discipline is not followed in all cases, the DSDS could probably deduce enough from the idents in the statement signatures of a program or data base and the information available in the identification system to be able to send a message to someone who would at least know how to forward it to a responsible party. 6c3b

The final issue which the DSDS must resolve is WHAT to say to the person (or other entity) which it has chosen to notify. In general, what the system needs to communicate to the debugger is sufficient information to identify the effective cause of the error. 6c4

To the extent that the system is able to identify the effective cause itself, the action required of the debugger(s) is more a matter of choice -- how (and whether) to "correct" the indicated bug -- than of analysis -- finding the bug in the first place. 6c4a

If the system is not confident that it has found the specific effective cause (and maybe in all cases) it must forward to the debugger(s) sufficient information about the total state of the system to permit them to perform the analysis themselves; if the system is one which permits "undoing" operations, then there should be enough information present to permit the debugger to "back up" the system and recreate the error condition itself. 6c4b

This is obviously an area which requires further thought, as it is intricately tied up with design and implementation considerations affecting the whole system in which the DSDS lives. 6c4c

COMPONENTS

In this section, I will attempt to summarize the preceding sections from a slightly different perspective by listing some of the components of a Dialog-Supported Debugging System, the functions of which have already been described. 7a

Program-initiated message delivery system 7b

Extension of the Dialog-Support System to permit programs as well as people to initiate dialog. 7b1

Functional role identification system 7c

Extension of the identification system to handle more flexibly the maintenance of structures of functional relationships and the assignment of roles to specific entities. Augmenting the entities handled to include programs and data bases as well as people and organizations. 7c1

Automatic dialog analysis, acknowledgement, and forwarding system 7d

A set of tools for augmenting the handling of messages sent to an addressee. Would have cataloging, content filtering, statistical analysis, automatic response generation, and selective forwarding capabilities. 7d1

Interactive bug reporting system 7e

The DSDS front end for interacting with users in the generation of bug reports. 7e1

Error-condition analysis system 7f

Tools for analysing the system environment at the time an error condition has been detected by program or user. Would provide mechanisms for helping to track down "effective causes" and decide who to report to. 7f1

Bug record system 7g

A set of management and organizing tools to to help debuggers keep track of the bug reports they receive, schedule bugs for subsequent fixing, maintain dialog with bug reporters to keep them informed, and record and document system changes. 7g1

Methodology of collaborative system development and debugging 7h

Part of the system Handbook describing the accepted discipline for collaborative system development and debugging. Would set forth the framework for Dialog-Supported Debugging and prescribe principles and standards for making it work. 7h1

15101 Distribution

Van De Riet, Edwin K. , Van Nouhuys, Dirk H. , Victor, Kenneth E. (Ken) , Wallace, Donald C. (Smokey) , Watson, Richard W. , Andrews, Don I. , Xerox PARC ,

Keeney, Marcia Lynn , Hoffman, Carol B. , Lee, Susan R. , Michael, Elizabeth K. , Dornbush, Charles F. , ARC, Guest O. , Feinler, Elizabeth J. (Jake) , Handbook, Augmentation Research , Kelley, Kirk E. , Meyer, N. Dean , Byrd, Kay F. , Prather, Ralph , White, James E. (Jim) , Vallee, Jacques F. , Kaye, Diane S. , Rech, Paul , Kudlick, Michael D. , Ferguson, Ferg R. , Lane, Linda L. , Auerbach, Marilyn F. , Bass, Walt , Engelbart, Douglas C. , Hardeman, Beauregard A. , Hardy, Martin E. , Hopper, J. D. , Irby, Charles H. , Jernigan, Mil E. , Lehtman, Harvey G. , North, Jeanne B. , Norton, James C. , Paxton, William H. , Peters, Jeffrey C. , Ratliff, Jake