JHB 6-NOV-74 17:51 24432

Directory Information Form

Information needed to set up new directories. 1

It would help us if you would use this form by copying it and then substituting the appropriate information for the characters. 1a

You may send the request(s) to Jim Norton (JCN), Jim Bair(JHB), Feedback (feedback), or for ARPA users, Susan Lee(SRL) via sndmsg or Sendmail. 1b

(FORM) 2

Directory name:	XXX
Ident:	III
Account:	same as architect
Password:	PPP
Disk pages:	300
Allocation group:	GGG
Default protection:	770000
Person's name:	NNN

2a

Ident Info: 2b

Address: 2b1

AAA 2b1a

Phone: 2b2

TTT 2b2a

JHB 6-NOV-74 17:51 24432

Directory Information Form

(J24432) 6-NOV-74 17:51;;;; Title: Author(s): James H. Bair/JHB;
Distribution: /KWAC([ACTION]) SRL([ACTION]) RLL([INFO-ONLY])
MEH([INFO-ONLY]) ; Sub-Collections: SRI-ARC KWAC; Clerk: JHB;
Origin: < BAIR, DIRMSG,NLS;5, >, 6-NOV-74 17:48 JHB ;;;;####;

KIRK 6-NOV-74 18:45 24433

Publish user-subsystem has Index and TOC, Not Format.

Check out the Publish user-subsystem and let me know what you think.
I erred by calling "Format" in a previous message. 1

KIRK 6-NOV-74 18:45 24433

Publish user-subsystem has Index and TOC. Not Format.

(J24433) 6-NOV-74 18:45;;;; Title: Author(s): Kirk E. Kelley/KIRK;
Distribution: /JHB([INFO-ONLY]) RWW([INFO-ONLY]) DVN([
INFO-ONLY]) JCN([INFO-ONLY]) RLL([INFO-ONLY]) ;
Sub-Collections: SRI-ARC; Updates Document(s): 24408; Clerk: KIRK;

KIRK 6-NOV-74 19:19 24434

Where do you want userprograms to be listed?

your sndmsg about not finding userprograms is disturbing. A list of userprograms is under the branch named userprograms under the branch named programs. It is linked to from the synonyms: usersubsystem, usersubsystems, content-analyzer, programing, userprogram, program, and etc.. Just exactly where were you wandering in Help? 1

6-NOV-74 1818-PST WATSON: Helpp and User Programs
Distribution: KELLEY, WEINBERG, vannouhuys
Received at: 6-NOV-74 18:18:29 2

In wandering in Help data base noticed that there is nowhere where user programs are listed. Probably should be . Thanks Dick 2a

KIRK 6-NOV-74 19:19 24434

Where do you want userprograms to be listed?

(J24434) 6-NOV-74 19:19;;;; Title: Author(s): Kirk E. Kelley/KIRK; Distribution: /RWW([INFO-ONLY]) DVN([INFO-ONLY]) POOH([INFO-ONLY]) ; Sub-Collections: SRI-ARC; Clerk: KIRK;

JCN RWW 7-NOV-74 10:56 24435

Procedure for Handling Phone Calls

There is presently a growing over load of work for Joan and Sandy and some change is needed in the handling of telephone calls. We have asked Joan and Sandy not to run out and try to search for people on each call unless the caller says it is urgent and instead to take the name and phone number, and any message and post it on the board. If you are expecting an important call you can tell them about it and ask them to find you if the call should come through. Most calls are not that urgent and a return call should be acceptable. Thanks for your cooperation. Dick, Jim

1

Procedure for Handling Phone Calls

(J24435) 7-NOV-74 10:56;;;; Title: Author(s): James C. Norton, Richard W. Watson/JCN RWW; Distribution: /SRI-ARC([INFO-ONLY]) ; Sub-Collections: SRI-ARC; Clerk: RWW;

JML 7-NOV-74 11:29 24436

Letter to A. Altree, Fireman's Fund Insurance, sending information about ARC and a lead to DIALOG at Lockheed

Augmentation Research Center
Stanford Research Institute
333 Ravenswood Avenue
Menlo Park, California 94025

Mr. Alan Altree
Fireman's Fund Insurance
4006 Ben Lomond
Palo Alto,
California 94306

Dear Mr. Altree:

From our telephone conversation on November 6th I wasn't able to sense the extent of your background in computer applications, but at any rate you will probably find that the concepts described in the enclosed paper are pretty "heady" stuff. 1

However, if you look closely, I think you'll find in sections 4c1 through 4c4 a fairly accurate description of the kinds of capabilities you will probably be looking for in your project of finding an on-line accessing system for insurance policies. 2

Probably you will want some means of making changes to these policies, either on- or off-line; a way to index them and organize them that is logical and easy to use; a means of generating hard-copy when needed, and so forth. 3

What we have at ARC is a state-of-the-art research group which is interested in collaborating with other groups who, while having specific computer system needs, are also interested in taking part in the on-going development of whole-picture workshop systems. 4

Generally those people who buy our services tend to be government or commercial groups who have a vested interest in the further development of computer-augmented research. 5

I suspect that within 10 or 15 years the capabilities of our system, so obviously useful in a situation like yours, will be readily available, to be bought on the marketplace as easily as other less all-encompassing systems are now. 6

Letter to A. Altree, Fireman's Fund Insurance, sending information about ARC and a lead to DIALOG at Lockheed

Meanwhile you have the job of finding a commercially available system for your needs. Other than just pointing you towards the vast diversity and complexity of systems presented in the computer research and EDP literature, I can give you one specific lead: 7

At Lockheed in Sunnyvale, the Information Systems Division, I believe, has an information retrieval system called DIALOG, and an on-line text editor, called AUTOTEXT, which are commercially available. I suggest you scout them out for further information. 8

Best of luck in your search. 9

Sincerely,

Jeanne M. Leavitt
Augmentation Research Center

Enclosure:

D. C. Engelbart, R. W. Watson, J. C. Norton, THE AUGMENTED KNOWLEDGE WORKSHOP, paper presented at the National Computer Conference, New York City, June 1973. (IJOURNAL, 14724,)

JML 7-NOV-74 11:29 24436

Letter to A. Altree, Fireman's Fund Insurance, sending information about ARC and a lead to DIALOG at Lockheed

(J24436) 7-NOV-74 11:29;;;; Title: Author(s): Jeanne M. Leavitt/JML; Distribution: /DCE([INFO-ONLY]) ; Sub-Collections: SRI-ARC; Clerk: JML; Origin: < LEAVITT, NEWTRY.NLS;3, >, 7-NOV-74 11:21 JML
;;;;####;

DVN 7-NOV-74 16:46 24437

One More thought about Journal Deliveryy

As a step toward (documentation, final,,6b6c4) and the rest of that plex, what the journal should do is enter in everyone's initial file an author,keword, and arrival data catalog of journal items sent to her or that she sent. 1

DVN 7-NOV-74 16:46 24437

One More thought about Journal Deliveryy

(J24437) 7-NOV-74 16:46;;;; Title: Author(s): Dirk H. Van Nouhuys/DVN; Distribution: /RLL([ACTION]) FDBK([ACTION]) JAKE([INFO-ONLY]) KEV([INFO-ONLY]) ; Sub-Collections: SRI-ARC; Clerk: DVN;

DVN 7-NOV-74 17:02 24438

Request to Journalize Draft on Journal System

Carlson for Lukasic (sp?) has asked for some information on the Journal. Among other things I would like to send him your draft (documentation,final,6b) as support. Why don't we journalize it, clearly marked DRAFT, so we get a nice, familiar format and can get at it easilly again, since I note it has been used this way a couple ot times before. 1

DVN 7-NOV-74 17:02 24438

Request to Journalize Draft on Journal System

(J24438) 7-NOV-74 17:02;;;; Title: Author(s): Dirk H. Van Nouhuys/DVN; Distribution: /JEW([ACTION]) ; Sub-Collections: SRI-ARC; Clerk: DVN;

KIRK 7-NOV-74 21:34 24439

My initial file went bad

Is it possible to re-send the journal items that have been sent to me between 6-NOV-74 17:00 and 7-NOV-74 21:30? 1

KIRK 7-NOV-74 21:34 24439

My initial file went bad

(J24439) 7-NOV-74 21:34;;;; Title: Author(s): Kirk E. Kelley/KIRK; Distribution: /JDH([ACTION]) ; Sub-Collections: SRI-ARC; Clerk: KIRK;

DVN 7-NOV-74 22:51 24440

Conversations about COM with George Lithograph

Following (journal,24317,) Ann Weiberg, Dean Meyer, Robert Bellville, and I met with Ron Feria. He gave us samples of the type faces available via the Singer 6000 from George. They have faces roughly equivalent to what we've been getting from DDSI and also a range of fancier faces. He assured us of their abilities and intent to try to work with us. We gave him a tape and our specs (journal,14093,) and told him to call back when his programming people had questions. 1

The following Monday Elizabeth returned from vacation. She knows George and is confident in them as printers and as business-like people, but doubted if they had the programming background to understand what we had given them. I called Nick Zsabo at Singer and suggested that we might have to talk to Singer people about programming. He indicated that they gave programming support to George and said a Richard Cox on their programming staff would call Elizabeth. He has not done so. 2

In the meantime Al Wilson, a programmer at George called and then visisted yesterday. For the most part he met with Elizabeth and Robert Bellville although Dirk van Nouhuys spoke with him as well. 3

This meeting was encouraging. The hardware appears to do everything the Comp 80 does. Their programmer has gone over our COM tape layout, and he says there will be no problem handling it. We are working together on a formatting (coding) scheme for intermixed line drawings. 4

They advertise overnight turn around time for plates. Time for actual prnting of n copies of a document depends of course on the volume. We have not yet discused price. 5

However like DDSI, George lacks the optics to make 48X reduced microfiche; we brought Airforce plans to use 48 X reduction to the attention of Al Wilson. 6

DVN 7-NOV-74 22:51 24440

Conversations about COM with George Lithograph

(J24440) 7-NOV-74 22:51;;;; Title: Author(s): Dirk H. Van Nouhuys/DVN; Distribution: /JOAN([ACTION] foor DPCS notebook) DPCS([INFO-ONLY]) LAC([INFO-ONLY]) DLS([INFO-ONLY]) ; Sub-Collections: DPCS SRI-ARC; Clerk: DVN;

JBP 8-NOV-74 08:48 24441

TIP Logon

What ever happened about arranging tip logon names for arc staff. 1

TIP Logon

(J24441) 8-NOV-74 08:48;;;; Title: Author(s): Jonathan B. Postel/JBP; Distribution: /RWW([ACTION]) JCN([ACTION]) ; Sub-Collections: SRI-ARC; Clerk: JBP;

JBP 8-NOV-74 08:56 24442

Terminal for postel

It might prove useful to privide me with a terminal at home now. I probably would not use it a great deal at first (while still living in this apartment) so would not expect sri to put in a seperate phone. When we do get moved into termite haven then i would want the terminal to have a seperate sri provided phone.

1

JBP 8-NOV-74 08:56 24442

Terminal for postel

(J24442) 8-NOV-74 08:56;;;; Title: Author(s): Jonathan B. Postel/JBP; Distribution: /RWW([ACTION]) ; Sub-Collections: SRI-ARC; Clerk: JBP;

Primer Feedback

We are still working on the primer and i is in a stage where things can be easily changed. If you can give us any input at all, I would greatly appreaciate it. I hope your trip is going well and I will see you when you get back. 1

Primer Feedback

(J24443) 8-NOV-74 09:51;;;; Title: Author(s): Anne Weinberg/POOH;
Distribution: /SRL([ACTION]) POOH([INFO-ONLY]) ;
Sub-Collections: SRI-ARC; Clerk: POOH;

Tenex Documentation

The following documentation can be ordered from BBN. Please let me know if you would like any of these manuals. I plan to place an order in the next week.

TENEX EXECUTIVE Manual
TENEX JSYS Manual
TENEX USER7s Guide
TENEX memos (1970)
TENEX Monitor Manual
PAL11X
BCPL
TENEX TECO

1

Tenex Documentation

(J24444) 8-NOV-74 10:08;;;; Title: "; Author(s): Anne Weinberg/POOH; Distribution: /SRI-ARC([ACTION]) ; Sub-Collections: SRI-ARC; Clerk: POOH;

Procedure and Forms for Identsystem Maintenance

If you can't live with any of this, lets get together and discuss

Procedure and Forms for Identsystem Maintenance

PROCEDURE 1

1. Peters will be responsible for setting up new directories per authorization of Norton or Bair. 1a

2. MLK will be responsible for adding new entries to ident system. Input may come from JCN JHB, JAKE, ACM, or clients. 1b

3. MLK will notify JAKE as to what new idents have been entered or changed for either individuals, groups, or organizations. 1c

4. JAKE and ACM will edit these (particularly for organization) and obtain missing information, if any. 1d

5. Changes can be made by either ACM or MLK, again with checking afterward by JAKE. This will give ACM a chance to learn all facets of the ident system and will give us more than one person who knows the system. 1e

6. Before the ARPAnet Directory goes out ACM will send out current listngs to each individual and a set of all listings for a given host to each principal investigator for verification. 1f

7. JAKE/ACM will run NIC program to produce individual listing on identfile and will also check items that 'fall through' the program to make sure no network people have been missed. 1g

8. JHB and JAKE will co-ordinate requests for idents by either filling out a form ourselves or by asking individual to submit a form (see below) with the proper information filled in. JHB will deal with architects and on utility to obtain proper information; JAKE will deal with Principal Investigators and individual at network hosts to obtain proper information. 1h

FORMS 2

INDIVIDUAL 2a

INFORMATION FOR INDIVIDUAL DIRECTORY 2a1

Directory Name: 2a1a

Account: 2a1b

Passwork: 2a1c

Alloted Disk Pages: 2a1d

Default Protecton 2a1e

INFORMATION FOR INDIVIDUAL IDENT ENTRY 2a2

Name:
Last First Middle Initial 2a2a

Home Organization (If host, use host name): 2a2b

Phone:
Area code Number Ext (if any) 2a2c

U.S. Mail Address:

2a2d

Network Mailbox:
USERNAME @ HOSTNAME 2a2e

NLS Mailbox: (SRI-ARC or OFFICE-1): 2a2f

Delivery (Journal or Network): 2a2g

Function: 2a2h

ORGANIZATION 2b

Name of Organization: 2b1

proposed Ident (if host, use host name): 2b2

U.S. Mail Address:

2b3

Network Mailbox:
USERNAME @ HOSTNAME 2b4

NLS Mailbox: (SRI-ARC or OFFICE-1): 2b5

Delivery (Network or Journal): 2b6

Phone:
Area code Number Ext (if any) 2b7

CO-Ordinator: 2b8

Membership (use NLS idents): 2b9

Type:
Server User Tip Assoc. Independent 2b10

SPECIAL-INTEREST GROUPS 2c

Name of Group: 2c1

Proposed Ident (if host, use host name): 2c2

Membership (use NLS idents): 2c3

CO-Ordinator: 2c4

U.S. Mail Address:

2c5

Network Mailbox:
USERNAME @ HOSTNAME 2c6

NLS Mailbox: (SRI-ARC or OFFICE-1): 2c7

Delivery (Network or Journal): 2c8

Function: 2c9

Procedure and Forms for Identsystem Maintenance

(J24445) 8-NOV-74 12:01;;;; Title: Author(s): Elizabeth J. (Jake) Feinler/JAKE; Distribution: /JHB([ACTION]) JCN([INFO-ONLY]) ACM([INFO-ONLY]) ; Sub-Collections: SRI-ARC; Clerk: JAKE;
Origin: < FEINLER, IDENTS.NLS;3, >, 8-NOV-74 11:58 JAKE ;;;;####;

RWW 8-NOV-74 12:34 24446

Conversation with Jack Goldberg relative to NSW

Jack Goldberg has been chating with Craig Fields about things that he might be able to do for him and Bill Carlson also. He wants to get into the NSW game somehow and is aware of not wanting to compete with us. Even if he does not compete in a task sense there is still possible competition for the funding pie. I spent an hour briefing him on what is happening, but am not quite sure how to handle coordination in talking with ARPA as he alredy is. Carlson is talking about giving him an additional 1/2 man year of funds under our contract to prepare some tools, he is building to run under NSW Ie use the frontend etc. We should chat about this. Dick 1

Conversation with Jack Goldberg relative to NSW

(J24446) 8-NOV-74 12:34;;;; Title: Author(s): Richard W. Watson/RWW; Distribution: /DCE([ACTION]) JCN([INFO-ONLY]) ; Sub-Collections: SRI-ARC; Clerk: RWW;

What I Think We Agreed to Do at NSW Meeting (are there other things let me know) 1

Protocols 1a

A Reviewable draft of File, RJE, Exec, TBH packages. 1a1

Reply to Black Boxes document indicating what combination of procedures supply which black box functions. 1a2

Possibly some revisions to PCP that I did not fully catch as I was out of the room. 1a3

Frontend 1b

Finish Our Design Documents 1b1

Decisions document relative to Frontend issues raised in CHI paper. 1b2

L 10 BCPL comparison 1b3

Some "help" capabilities for Works Manager for file name or othr parameter completion, details not clear as I was not in discussion. 1b4

NLS 1c

Finish our tasks to do document. 1c1

Double check need for intermediate way to set up OP Directives 1c2

Provide and initial NSW NLS based Help and User Profile creation tool. 1c3

Other 1d

We did not agree to it but I think JEW, JBP and CHI should produce a process structure model for whole NSW that deals with issues discussed in Thurs scenario discussions. 1d1

What I Think We Agreed to Do at NSW Meeting

(J24447) 8-NOV-74 13:08;;;; Title: Author(s): Richard W. Watson/RWW; Distribution: /JBP([ACTION]) NPG([ACTION]) DCE([INFO-ONLY]) JCN([INFO-ONLY]) ; Sub-Collections: SRI-ARC NPG; Clerk: RWW;

ARC Journal: ROUGH DRAFT of Report Chapter

Our Conception of Dialog Support 1

RECORDED DIALOG 1a

One of the prime objectives of the augmentation system developed at ARC is to aid collaborating knowledge workers by providing flexible computer tools and methodology for communicating with one another. We collectively refer to such tools and methodology as a Dialog Support System (DSS). Its primary task is to provide mechanisms for transmitting on-line messages and documents between users. However, for large projects or those about which some larger community of users must remain informed, the dialog soon becomes unmanageable without additional computer aids. ARC's DSS therefore: 1a1

1) permanently records (copies to read-only storage),
2) numbers (assigns a unique accession, or catalog number),
3) and catalogs (records author, title, number, and location) 1a1a

each piece of dialog -- for later consultation, for reference by later documents, and for examination by interested bystanders. 1a2

THE JOURNAL 1b

ARC's DSS is implemented as a set of computer processes called the Journal, consisting of a foreground subsystem which interacts with the user and provides primitives for entering a message or document in the Journal (with title, author and other information), reserving catalog numbers, and so forth, and a background process that further processes submission requests and delivers mail to the addressees indicated by the author. The Journal is supported by several additional systems: an Identification System responsible for maintaining information about users -- their location, group memberships, phone numbers, and so forth -- and a Number System responsible for keeping track of which catalog numbers have been assigned and to whom, and which are available for future assignment. 1b1

Since its implementation in April of 1971, the Journal has been heavily used (containing at present over 10,000 messages and documents), initially by the ARC staff, then by a larger user community with network access to ARC's computer facility, and most recently by commercial and government users of a second computer facility operated for ARC. The Journal has evolved as a result of our experience with it and in response to the increased demands placed upon it by its growing user base. This report describes that experience and evolution. 1b2

Our Initial Implementation 2

THE ARC/NLS ENVIRONMENT 2a

ARC's DSS resides on a heavily loaded Digital Equipment Corporation (DEC) PDP-10 running Bolt, Beranak, and Newman's (BBN) Tenex operating system. Tenex provides a time-sharing environment in which 10-20 users independently interact with any of a variety of applications packages called "subsystems". ARC's PDP-10 is devoted almost exclusively to providing access to a single subsystem, ARC's On-line System (NLS) (13041,), a comprehensive system of tools for manipulating structured text. 2a1

NLS provides a very general set of primitives for manipulating and viewing tree-structured text files. Commands are provided for manipulating the tree's structure, e.g., for adding nodes called "statements" to the tree, for deleting single statements or whole branches of the tree, for moving or copying a subset of the tree from one location to another, and so forth. 2a2

In order to maintain flexibility in the first implementation and to facilitate maintainence of the system, NLS text files were consistently used in implementing the Journal, Identification, and Number Systems' principal data bases, as well as for catalogs, indices, and a variety of internal, inter-process communication files. 2a3

STRUCTURE 2b

The Journal 2b1

The Journal System is a set of procedures that run in both foreground and background modes to maintain a data base of recorded documents, and to distribute them to specified addressees. 2b1a

Larger Journal documents are stored as separate files in a set of system directories. Short documents, called "messages", given special treatment in the interests of economical storage, are stored in a set of (currently about 20) files, several hundred to a file. Whenever a document remains unreferenced for a month, it is archived to magnetic tape by Tenex, and its on-line storage released for other use. Although over 10,000 items have been journalized on the PDP-10 since April of 1971, most have long ago been archived and therefore do not occupy on-line storage, except when brought back for reexamination. 2b1b

The Journal maintains a system catalog of all recorded

documents, implemented as a set of (currently five) on-line files. The catalog contains information used by NLS to locate a Journal item given its catalog number, as well as information used by stand-alone programs to produce non-system catalogs and indices (by author, titleword, and number). 2b1c

Journal mail addressed to a particular user is delivered in one or both of two delivery modes, On-Line and Hardcopy. The delivery parameters are selected by the addressee and maintained by the Identification System. A document's author need know nothing about the delivery mode of its addressees. 2b1d

ON-LINE DELIVERY 2b1d1

Regular users of NLS normally receive on-line delivery of all their Journal mail. Each item is placed by the Journal in a special NLS file called the user's "initial" file (so named because the file's name is the user's ident, which in turn is usually his initials). For convenience, this file is automatically loaded for the user when he enters NLS. The text of short messages is delivered to the user in its entirety. For longer items, only a citation giving the document's author, title, and date, and a convenient, machine-readable pointer (called a "link") to the text of the document are delivered. 2b1d1a

HARDCOPY DELIVERY 2b1d2

Hardcopy line printer output is sent by U.S. Mail to users who never or only infrequently use NLS or who, for one reason or another, want it in place of or in addition to on-line delvery. A substantial amount of clerical support is required to support hardcopy delivery. 2b1d2a

The Journal maintains information about on-going distribution operations in a single, NLS file, used also as a vehicle for communication between the submission and distribution components of the background system. 2b1e

The Identification System 2b2

The Identification System is a set of procedures that maintain a large data base, implemented as a single, very large NLS file, containing information about individuals, groups of individuals, and organizations (each of which is

assigned a unique name called an "ident"). Various information fields are maintained for each ident, and procedures are provided for manipulating each field. 2b2a

The Identification System includes an NLS subsystem that permits users to interrogate and modify the data base themselves, subject to the appropriate access controls. 2b2b

Because of the data base's size, and because updating the data base involves creation of a new version of the file (requiring about 30 seconds or more of real time on a loaded system), all of the changes for a particular ident are collected from the user before the file is updated. 2b2b1

The Number System 2b3

The Number System is a set of procedures that manage a data base, implemented essentially as a single NLS file, containing information about the assignment of catalog numbers to Journal documents. The data base contains 2b3a

1) a number of blocks of numbers available for assignment. 2b3a1

2) a list of assigned numbers (either recently used, assigned but as yet unused, or in the process of being used) and for each the date and time of assignment and the ident(s) of the user(s) to whom it was assigned. 2b3a2

It is often useful to know in advance what number will be assigned by the system to a particular item. This is necessary, for example, to create a set of documents that internally reference one another. A catalog number may therefore be reserved for later submission, or "preassigned". 2b3b

The RFC number system, a separate special-purpose number system patterned after the master system (and thus able to use most of the same primitives), was implemented at the request of an informal group of network protocol developers. An item may have an RFC number in addition to the master catalog number. 2b3c

EXPERIENCE AND PROBLEMS 2c

A number of problems with the initial Journal implementation have been encountered and attacked. Some of the major problems are described below. 2c1

Excessive real-time required for submission 2c1a

In the initial implementation, the entire submission process, with the exception of delivery, was performed in the foreground and therefore kept the user from other work for what often, given the system load, proved to be an inordinate amount of time. In an attempt to alleviate this problem, the submission mechanism was restructured, and all manipulation of catalog, distribution, and storage files deferred to the background process. 2c1a1

A special system directory was established for queuing submission requests for the background process, that now goes through two distinct phases. First, all queued submissions are processed: numbers are assigned where necessary, the document is stored in the appropriate message or separate file in the appropriate system directory, the document is cataloged, and a distribution request is queued. And second, whatever distribution requests have accumulated are processed, one individual addressee at a time. 2c1a2

To further reduce the amount of processing that must take place in the foreground, a form of submission is permitted in which the task of assigning a catalog number is deferred to the background process. Deferred submission is the default, and most submissions are therefore of this type. Since deferred submission does not require write access to any system files, a user can submit an item in this mode at any time, regardless of the state of the Journal or Number System files. 2c1a3

Background delivery degraded system performance 2c1b

The Journal background process has proven to be very expensive to run, and often has had a detrimental effect upon the responsiveness of the system as viewed by its interactive users. We've experimentally varied the frequency with which the background process runs (and thus with which mail is delivered), from once per day initially, to its current frequency of once every hour. 2c1b1

The background process now periodically checks the load average (the Tenex monitor's measure of system demand) and suspends processing if it is above some predetermined cut-off value. Processing is resumed only when the load average drops sufficiently. The check is performed at a point in the process when the system files are consistent

and least vulnerable to a crash. Between these check points, the process runs at high priority. 2c1b2

The benefits of this strategy are threefold: the background process does not add appreciably to the system load when it's already high; it can exploit slack times throughout the day; and since the probability of a crash increases with system load, the Journal and Number System files are usually in a relatively invulnerable state when a crash occurs. 2c1b3

Data bases vulnerable to system failures 2c1c

A very serious problem of the initial Journal implementation was the vulnerability of the various system files to hardware (especially disk) problems, monitor crashes, and exhausted disk storage. The processing of hardcopy output, besides being time consuming, was similarly vulnerable to both software and hardware failures. 2c1c1

The danger of losing system files because of lack of disk storage has been greatly reduced by also checking for available disk space at the same time the load average is checked. Processing is terminated until the next hour if space is too low. This strategy prevents loosing a system file due to exhausted disk space during a file update. 2c1c2

A number of problems associated with the processing of hardcopy output have been largely eliminated. A variety of monitor bugs have been fixed or avoided. The bulk of the processing is done during the evening or early morning hours. Because of the volume of hardcopy output produced by the Journal, the print requests were first placed on magnetic tape and printed on an IBM 360 system elsewhere at SRI, and finally contracted outside of SRI. Network delivery, described in the next section, has, on the other hand, drastically reduced the volume of hardcopy produced, and thus recently permitted us to resume printing on our own system at OFFICE-1. 2c1c3

Extensions for a Network Environment 3

THE ARPANET ENVIRONMENT 3a

In July of 1970, ARC's PDP-10 became part of the ARPANET, now an international network of large-scale computer facilities called "hosts" linked by 50 kb communication lines. Once the

lowest level, inter-machine communication protocol was developed, the central task was to design and implement the software protocols required for general, inter-process communication and other, more specialized exchanges. This task was undertaken by an informal group of geographically separated systems programmers called the Network Working Group (NWG). 3a1

In early 1969, ARC had offered to serve as the Network Information Center. As soon as hardware connections were made and protocol development reached a stage sufficient to permit simple, teletype-like use of a remote time-sharing system, ARC began to provide dialog support for the NWG via the Journal. 3a2

JOURNAL CHANGES TO SUPPORT THE NETWORK 3b

At first, the Network user used the Journal in nearly the same manner as a local user. Like local users, he had to login to the ARC system and use NLS to compose and journalize a document. But unlike most local users, he received hardcopy, rather than on-line delivery of his Journal mail. When ARPANET protocols developed to the point of permitting the transmission of text files and mail to users at remote hosts via the Network itself, the Journal was modified to utilize this new capability. 3b1

Network delivery 3b2

The File Transfer Protocol (FTP) (17759,) devised by the NWG permits the transmission of text to a named "mailbox" at a remote host. For purposes of receiving mail, therefore, each Network user has a "network address" consisting of a host name and a mailbox name. To exploit this new Network capability, we added a third, "network" delivery mode to the existing on-line and hardcopy modes, storing a network address in the ident file for each Network user. A Network user can thus take delivery of all Journal mail addressed to him, in his own system, simply by storing the appropriate delivery parameters in the Identification System. 3b2a

Rather than deliver extremely long documents via the Network in their entirety, we made the same size distinction for network delivery as for on-line delivery, sending only a citation for long documents. We modified the FTP software supplied by BBN to recognize a distinctive pathname (that the Journal provides with the delivered citation) that, when used to retrieve Journal documents, invokes a conversion of the tree-structured document to sequential form before transmission through the Network. A Network user can thus

explicitly retrieve the full text of any Journal document sent to him. 3b2b

Network submission 3b3

The fact that the Network user had to explicitly connect to and login at ARC's pDP-10 to enter a document into the Journal, and that he had to compose the document using NLS, complicated life for some users, forcing them to learn the details of NLS, in which some had only one, specialized interest. 3b3a

To alleviate this problem, we implemented a facility that permits users to journalize documents composed via their local editor without explicitly connecting to the ARC system or logging in, and without any knowledge of the NLS command language. We did this by further modifying BBN's FTP software to recognize a special mailbox name of the form "authors/addressees" and to interpret it, in the context of a mail delivery, as a Journal submission. The ident lists "authors" and "addressees" are verified by NLS, running beneath the FTP program in an inferior fork. If the ident lists are found correct, the "mail" is immediately journalized. Thus the remote user can journalize a document using the normal, Network mail facility provided by his system. 3b3b

EXPERIENCE AND PROBLEMS 3c

The Journal's Network submission and delivery facilities have been in operation for over a year now. The latter has suffered from a few, relatively minor problems. Network addresses, for example, are not well understood by some users who, in attempting to modify them themselves, have frequently modified them incorrectly. In such cases, delivery of the user's mail is prevented until the error is discovered and corrected by ARC personnel. Because of this, almost all Identification changes are now done by ARC staff. Many users are unwilling to explicitly retrieve the text of long documents for which they are sent only a citation, even though the retrieval process is straightforward, even automatable. 3c1

The submission facility suffers from more severe problems, one of which is that the ident verification and journalization processes are very time consuming and must be completed before the user's request is acknowledged and he is "set free". A more satisfactory strategy would be to queue the request and acknowledge it immediately, releasing the user for other work,

and then to perform the expensive processes in background mode, with a Network message sent to the author in case of failure. 3c2

A second problem is that the conversion that the Journal must make between the sequential text file presented by the user and the tree-structured NLS file required by the Journal is often unsatisfactory to the user. We believe this to be a very difficult problem to solve, one perhaps best handled by permitting the inclusion of sequential files in the Journal data base, thereby eliminating the need for conversion. 3c3

A final problem is the inadequacy of the mail subset of the FTP protocol, which makes it difficult or impossible for the user to transmit any of the optional parameters supported by the Journal, and which forces the user interface to remain somewhat artificial. ARC has proposed a separate mail protocol (17140,), but no protocol development is being carried out in that area at present. 3c4

Extension to a Dual-Site System 4

THE SRI-ARC/UTILITY ENVIRONMENT 4a

In January of 1974, ARC began operation of a second, "Utility" PDP-10 system we call OFFICE-1 to provide NLS support in a stable environment to what has proved to be an ever-growing clientele. The facility is operated for ARC by Tymshare, Inc, from Cupertino, California. Like ARC's own PDP-10, OFFICE-1 is connected to the ARPANET, through which most of its users gain access to it. The Utility's software configuration is essentially identical to ARC's, providing the full range of NLS service to its users. One such service is, of course, the DSS. 4a1

In providing Journal service from the Utility, we decided to include that second system within the domain of what is conceptually a single Journal spanning both the ARC and Utility machines. That is, rather than simply replicate the software, thereby creating a second, independent system, we decided to couple the two DSS systems, making all items journalized from either system available at both and addressable to users resident on either machine. Thus, for example, we employ a single Ident File, but maintain it in duplicate. 4a2

STRUCTURAL CHANGES 4b

In implementing a dual-host Journal, we were somewhat pressed for time and therefore decided to design and implement a interim system and later replace it with a more efficient and carefully thought out implementation. 4b1

The interim dual-host journal we decided upon involves duplicate Journal, Identification, and Number Systems, cognizant of each other at only a few points in the code. The two systems communicate with one another through the ARPANET via FTP. We implemented a special, assembly-language module to perform the FTP operations on NLS's behalf, since the corresponding FTP software provided by BBN is neither designed to be called by another program (since it's implemented as an interactive subsystem) nor structured in such a way that the relevant subroutines can be easily extracted. The portion of BBN's FTP software that was retained has been modified to deal more satisfactorily with NLS files, which have blank spots in their address space. 4b2

Two Journal Systems 4b3

Each submission request, regardless of its source, is fully processed by the Journal System on each machine. Each system's journal catalog and document files, though in a sense maintained independently, are always identical (neglecting the obvious time lag). To avoid duplicate delivery of each Journal item, as would naturally occur as a consequence of duplicating the submission request, we partitioned the idents, assigning responsibility for delivering mail to any particular user to (in most cases) just one of the two systems -- the one on which the user does most of his work. 4b3a

Submission requests are duplicated in the following manner. The background process on each system, before processing recent submissions, moves any files in the other host's special communication directory (OUTJOURNAL) to a local submission queue directory (TEJOURNAL), thus adding them to the list of local submissions to be processed. Then, in processing that list, a copy of each submission request, except those obtained from the second host, is queued for the other system in the local communication directory (OUTJOURNAL again). 4b3b

Two Identification Systems 4b4

To simplify the task of uniting the two Identification Systems, we bypassed the problem entirely by permitting additions and modifications from only one machine. The other machine is periodically sent an updated copy of the entire data base. 4b4a

Two Number Systems 4b5

The two Number Systems function independently, each assigning catalog numbers from a separate block. Numbers preassigned on one machine must be used on that machine, and the RFC Number system is available on only one machine. 4b5a

EXPERIENCE AND PROBLEMS 4c

Aside from the obvious inefficiency of duplicating each submission on the remote machine even though the item may be of only local interest, there have been no serious problems with our interim implementation. 4c1

An occasional asynchrony problem arises as a result of the time delay between an addition or modification to the ident file and receipt of the modified version of the data base at the second machine. For instance, an ident could be added to the Identification System, a Journal item sent to him from that machine (which already knows of his existence), and the item could reach the remote system via FTP before that system becomes aware of his addition to the system, causing an error in the remote system's Journal delivery function. 4c2

The most common problem with the dual-host system is Network transmission errors during file transfers. Such failures cause the item being transmitted to be delayed until an operator finds the file in an unusual state on the source machine. He must then check the destination system to verify that the file has not in fact arrived (which is the usual case), and then requeue it for transmission. Since occasional Network failures are inevitable, we are attempting to enhance the performance of the dual-host system by automating the detection and requeuing process. 4c3

The redundancy of information within the dual-host system is occasionally useful for reconstructing data lost due to a malfunction of the file system. A backup of the file system recently experienced by the Utility cost no more than reconstruction time; no Journal files were lost. 4c4

Private Dialog 5

COMING TO GRIPS WITH THE PROBLEM 5a

From the outset, one of the design goals for the Journal has been to provide an atmosphere in which memos, formal design documents, proposals, and other items, once published, would thereafter be readily accessible to anyone who cared to consult them. Author and subject indices are periodically produced and anyone, whether an active participant in the dialog or not, can

therefore browse through the list of items authored by a particlar individual or written on a particular subject, skimming or reading in full any items that look useful or appealing to him. 5a1

This model of dialog was appropriate for the system's initial user community, ARC itself, where subgroups working on highly inter-related tasks must keep abreast of one another's activity. As the Journal's user community grew to encompass researchers throughout the ARPANET, the model remained for the most part appropriate. Again the participants were engaged in separate but inter-related subtasks of a single, large project (i.e., ARPANET protocol design and implementation), and each working group had legitimate (and often vital) interest in the work of the others. But with the extension of the Journal to a dual-host system, a new class of users became involved. Many Utility users, though anxious to use the Journal as a dialog support aid, were not at all anxious to have all of their dialog (including, perhaps, personal correspondence, new product information, and so forth) accessible to the general public. Thus ARC was compelled to address itself to the problems of non-public, or private dialog, and to provide support for it through the Journal. 5a2

CHANGES TO THE JOURNAL 5b

What follows is a brief discussion of the more fundamental implementation problems which we encountered in tackling this problem; the reader is referred to (22911,) for a more detailed statement of the Journal changes made. 5b1

The Journal must ask itself three questions in establishing a user's right to view a recorded document: 5b2

1) Who is it requesting access to the document? 5b2a

2) Has he explicitly been granted access to the document? 5b2b

3) Is he a member of any group (perhaps by way of one of more levels of indirection) that has been granted access to the document? 5b2c

Who is the requestor? 5b3

The Journal has always tolerated imposters, simply accepting the user's word for the ident he declares at login to be his. It has done so because it could afford to, and because it was difficult to do otherwise. 5b3a

Access to a user's personal files is controlled by the monitor, and all system files (i.e, Journal documents) were accessible to everyone. The only thing which hinged on the ident claimed by the user was the authorship of items he journalized during the session. 5b3a1

Since the Journal designates users by ident, rather than by directory name, and since elements of the two name spaces cannot, in general, be placed in one-to-one correspondence (several users, each with an ident, often sharing a single directory), the monitor's login identity check was of little use as it stood. 5b3a2

Rather than significantly perturb the Tenex login procedure, we adopted the following strategy. 5b3b

For those users who have personal directories, we constructed a system data base giving ident as a function of directory. Tenex was modified to infer the user's ident from his stated directory name (which, of course, had to be accompanied by the appropriate password) at login, using the data base, and to store it in a read-only, job-global cell for subsequent interrogation by NLS. 5b3b1

For those users who share a directory, we placed opposite the directory name in the data base the idents of the users who use the directory. When Tenex encounters such a user at login, it interrogates him for his ident, accepting only one that appears in the list. 5b3b2

Thus, those users who are assigned a personal directory, and who login only under that directory, are completely protected by the system (i.e., they cannot be impersonated), while those who work in a community directory, are less fully protected, since they can be impersonated by any other member of the directory community. We are encouraging user organizations to set up separate directories for each user. 5b3c

Has the requestor been granted access to the document? 5b4

We have defined two classes of Journal items: private and public. Whenever a document is entered in the Journal, its author can select the class most appropriate, with public being the default. Private documents are defined to be readable only by the clerk, an author, or a distributee. That list of idents, including in general those both of individuals and groups, is stored as text in the first statement of the file which ultimately holds the document in

read-only storage. Whenever a user attempts to load the file, the list is consulted, and if the requestor's ident appears in it, his request for the document is honored. 5b4a

Has he been granted access by implication? 5b5

Since authors and distributees may be groups of people (or other groups), as well as individuals, the access list for a private document in general contains group, as well as individual idents. A user who requests access to a private document may therefore have legitimate access to it by virtue of his membership in a group, without his individual ident appearing explicitly in the access list. Because group idents are used heaviliy is this way, we were compelled to provide efficient means for verifying an ident's IMPLICIT appearance in an access list. 5b5a

To this end, the Identification System was modified to maintain back links, as well as forward links between each group ident and the idents of its members. That is, not only is a membership list maintained for each group ident, but in addition, now, a group list is maintained for each individual or group ident, specifying the list of groups of which the ident is a member. 5b5b

The logged-in user's group list is loaded by NLS once per session, and by a simple search of that list, most instances of legitimate access attempts to private documents can be identified. For those cases in which the user's claim to a document is more complicated (e.g., requestor A is a member of group B that is a member of group C, that appears in the access list), the Identification System is consulted and its data base examined more thoroughly. 5b5c

EXPERIENCE AND PROBLEMS 5c

The private dialog feature of the Journal has been in advertised use for only a few months, and hence any in depth attempt to evaluate its performance or use would be premature. The areas in which effects are most likely to be expected are those involving intimate collaboration between users. It's long been common practice, for example, for cooperating users to impersonate one another to get at a file which, though necessarly residing in one particular directory, is in reality a joint file. In implementing private dialog, we've necessarily restricted such practices, and the result will probably be the design and implementation of more formal methods for accomplishing such shared tasks. 5c1

Our Thinking About a General, Multi-Site System 6

MOTIVATION 6a

Recognizing the immediate need to provide dialog support for Utility users, and recognizing also that the implementation of an efficient dual-host dialog support system would require significantly more than simple modification of the existing, single-host system, we elected to make the short-term modifications described earlier and then to begin design work on a general, multi-host system to be distributed on an arbitrary number of ARPANET host systems. 6a1

The implementation of such a system would involve a complete rewriting of the present Journal, Number, and Identification Systems. Furthermore, we expect that the new DSS will in many ways be a different system, one in which many of the basic concepts of the previous system find a place, but also one in which new concepts appear. 6a2

DESIGN GOALS 6b

In designing a Multi-Host Journal system (MHJS), we had a number of goals in mind, the first necessarily being modularity: 6b1

Modularity 6b1a

We envision a system composed of modules, each providing some specialized service to the others, or to the end user, and which together comprise a coherent system. 6b1a1

Each module implements a set of primitives whose syntax and basic function are to be standardized, but whose internal workings would be left unspecified by the design (within certain broad constraints), being dependent upon the implementation machine, and the particular role that the module is to play within the System as a whole. 6b1a2

Reconfigurability 6b1b

The MHJS must be reconfigurable. Although the design suggests in broad terms the manner in which the System is to be constructed from its component modules, the design does no more than specify a family of MHJS's from which a particular configuration can be selected (in the same way that a computer system manufacturer provides a set of hardware modules (disk drives, CPUs, etc.) from which the customer configures his particular system). 6b1b1

The design specifies a small set of module types, each of which is replicated in appropriate numbers for a particular system configuration. 6b1b2

The MHJS must be reconfigured, for example, to accommodate the addition of new hosts to the system, or it might be reconfigured to place an instance of a frequently used module closer to a population center, or for any of a variety of other reasons. 6b1b3

Optimum Data Base Distribution 6b1c

It is, of course, more expensive to manipulate remote data bases than local ones; sometimes it's impossible (e.g., when the remote host is down). The MHJS, therefore, must attempt to reduce the frequency with which remote data bases must be dealt with by replicating portions of them in centers of user population and message traffic. 6b1c1

Uniform and Consistently-Applied Access Controls 6b1d

The MHJS must recognize the existence of private information of every type (documents, catalogs, idents, etc.) and provide the access controls necessary to protect it, providing for private dialog of a much more flexible nature than that described in the preceeding section. 6b1d1

With these goals in mind, then, we began designing a Multi-Host Journal System. Some of the more important concepts we came up with are described below; the reader is referred to (23144,) for a more complete discussion. 6b2

SOME IMPORTANT CONCEPTS 6c

Isolating the Recording, Cataloging, and Distribution Functions 6c1

The original Journal implemented a single user primitive we called "submit" which records, catalogs, and distributes a document. We considered that primitive fundamental to dialog support, and the vision of it colored our thinking about the Journal's internal structure. We've since learned that the sub-primitives from which submit is constructed are also of interest to the user. 6c1a

For example, we've found it useful to be able to distribute a previously submitted document to additional users, an operation that we've implemented and call

"secondary distribution" (even the name reflecting our bias toward "submit"). We now recognize, further, the need to be able to distribute a document without recording it at all, a facility that the present Journal still does not offer. And we recognize the cataloging sub-function of "submit" to be a more generally useful tool, applicable, for example, to personal as well as system data bases. 6c1a1

Access Controls 6c2

We decided from the outset of the design to implement flexible access controls throughout the MHJS, applying them not only to documents, but to data elements of all types -- catalogs, idents, and so forth. Controlling access to a data element consists of specifying, when the data element is created, the list of individual and/or group idents granted access to it, and then limiting access to members of that list. 6c2a

This is the same kind of access control now implemented in the present Journal, as we've already described, and is by far the most satisfactory type of which we know. In the MHJS, we've taken the additional (and natural) step of assigning passwords to idents, and requiring their use, as a means of verifying the user's identity. 6c2b

Catalog Number Assignment 6c3

The present Journal assigns every recorded document a unique identifier called a catalog number by which the document can be referenced or retrieved. Since the MHJS is conceptually a single Journal, we must somehow maintain uniqueness in catalog number assignment, while yet hopefully making the assignment process reasonably efficient and reasonably insensitive to host failures. These requirements preclude the simplest implementation, i.e. assignment of numbers by a single module at a single host. 6c3a

The approach we think most satisfactory is to station several instances of a module we've called the Number Vendor at strategic points about the system. Each additional Number Vendor, assuming it resides on a different host, increases the probability of a user's being able to obtain a catalog number when he wants it, as well as reducing the overhead (by placing the source closer to him). 6c3b

At any point in time, each Number Vendor owns a subset of the universe of catalog numbers from which it can satisfy

user requests. A Number Vendor may only assign catalog numbers that it itself has been assigned by another Number Vendor, except for one special "root" number vendor assigned initial possession of the entire name space. 6c3c

Number Vendors might be stationed throughout the MHJS, each with responsibility for servicing a segment of the user population, and each replenishing its number supply, when it nears bottom, from the root vendor. This strategy permits a form of number assignment that is both efficient and insensitive to the host failures that periodically make the root Number Vendor inaccessible. 6c3d

Publishing a Document 6c4

In our design of a MHJS, we've made central a concept that is only given lip service in the present Journal, that of subcollections. A subcollection is a subset of all recorded documents, each of whose members shares some common attribute, e.g. author, subject, and so forth. A single document may be assigned to zero or more subcollections, either explicitly by the author, or implicity by the system. Although hardcopy subcollection catalogs can be generated, the Journal maintains no on-line subcollection catalogs, thus severely limiting the utility of the concept in its present implementation. 6c4a

A major concern of the MHJS is to provide specialized marketplaces in which documents can be exchanged. Such a marketplace is called a "forum", and one speaks of "publishing" a document in a forum. In the MHJS we've thus placed great stress on the concept of allying a recorded document with other documents related to it (i.e., placing it in a subcollection), relegating the concept of simply recording a document to a less central role. 6c4b

Users with interest in a particular forum can formally declare that interest, and, subject to appropriate access controls and accounting disciplines, become "subscribers" of it, thereafter automatically receiving an announcement of each new document published. The prime responsibility of the Publisher, the module which implements a forum, is therefore to catalog each document as it is contributed, and send a copy of the catalog entry (giving the document's author, title, date of publication, etc.) to each of its subscribers. We've thus given the old concept of subcollections an active, rather than passive character, with the system notifying interested users as new documents are made available. 6c4c

Maintaining Networks of Documents 6c5

For reasons of efficiency and reliability, it is necessary to permit an arbitrary number of physical copies of a document to exist simultaneously within the MHJS. Each additional copy, assuming it is created on a different host, increases the probability of a user's being able to retrieve the document when he wants it. A retrieval request can be satisfied most quickly, of course, if a copy of the requested document happens to exist on the user's own host already. The system might therefore create a copy of the document at each major population center, anticipating a rash of retrieval requests; and then delete the copies a month later, once the period of peak demand has passed. 6c5a

Access to a document and all its copies is uniformly controlled on the basis of access lists assigned by the author. A user, for example, cannot read a document unless the author granted him read access to it. The copying of documents, however, is a system function designed to promote efficiency and is therefore unhindered by access controls. 6c5b

Each recorded document within the MHJS is therefore implemented as a network of copies whose topology is a dynamic characteristic of the system and changes with such things as the frequency with which it is referenced. The system keeps track of the various copies of a document, and can thus direct the curious user to the nearest one. 6c5c

Distributing Information About Users and Modules 6c6

A need that pervades the MHJS, even more so than in the present Journal, is that of swift access to information about users of the system. In the present system the data base is called the Ident File and describes the users and user groups known to the system. To implement the access controls that the MHJS seeks to maintain throughout, both human users and system modules are assigned idents. Group idents are very heavily used, being extremely convenient for implementing access lists for the various data bases within the system. 6c6a

For reasons of efficiency and reliability, it is highly desirable to maintain copies of subsets of the Ident File at various locations within the system, each under the control of a module called a Registrar. An ident can be known to an arbitrary number of Registrars, and that particular set of Registrars is called the ident's "domain". Information about the ident can be obtained from any Registrar in its

domain. Modifications to an ident are relayed to all Registrars affected, by the Registrar that receives the modification request. 6c6b

The Registrar turns out to be the workhorse of the MHJS, and its importance cannot be underestimated. In designing the MHJS we discovered that: 6c6c

1) Virtually every system module must deal with incidental data bases which are lists of user/program names (e.g., access lists), and each must provide mechanisms for retrieving and modifying them. 6c6c1

2) System modules can be relieved of a significant burden by providing a specialized module (the Registrar) whose function is to provide the primitives required to manipulate these data bases. 6c6c2

3) Furthermore, the lists then become accessible from any one of an arbitrarily large set of Registrars (the group ident's domain), since the Registrar already implements the required broadcast facility. 6c6c3

4) Since the existence of a document's read access list (for example) implies the existence of the document itself, whether or not a document exists can be determined by consulting the nearest Registrar. 6c6c4

5) Race conditions associated with the creation of a document (e.g., two users attempting to create a document with the same catalog number simultaneously at two different points in the system), for example, can be arbitrated by the use of locking mechanisms implemented by the Registrars. 6c6c5

Conclusion 7

Having made heavy and continuous use of the Journal for over three years now, ARC has found it to be a powerful dialog support tool for knowledge workers. 7a

During the course of its use, the Journal has been substantially modified to increase its efficiency, extend its geographical reach, and provide the new features we've discovered to be important. Initially an experimental system supporting a fairly small number of geographically concentrated researchers, it now supports a large, geographically distributed user community linked by the ARPANET. Initially a software system implemented on a single computer, it now operates on a pair of PDP-10 systems

linked by the Network, and design work has been done for a general, multi-host system. Initially exclusively a forum for public dialog, it now supports private communication as well. 7b

The Journal will further evolve and new features will be implemented and experimented with as we continue to gain experience in the dialog support field. 7c

ARC Journal: ROUGH DRAFT of Report Chapter

(J24448) 8-NOV-74 13:49;;;; Title: Author(s): James E. (Jim) White/JEW; Distribution: /DVN([INFO-ONLY]) RWW([INFO-ONLY] I journalized this since we seem to need formatted copies from time to times) ; Sub-Collections: SRI-ARC; Clerk: DVN;

TENEX Documents

I would like to have a complete set, i think i have the BCPL and TECO manuals so please include in the order the EXEC, JSYS, USERS, manuals, the others (memos Montior, and PAL11X) i guess i could do by checking with an ARC reference set,
--jon.

1

TENEX Documents

(J24449) 8-NOV-74 14:53;;;; Title: Author(s): Jonathan B. Postel/JBP; Distribution: /POOH([ACTION]) ; Sub-Collections: SRI-ARC; Clerk: JBP;

ARC Dialog Support: ROUGH DRAFT for a Report Chapter

This was journalized because we seemed to need formatted copies from time to time.

Dialog Support: the NLS Journal, Identification, and Number Systems 1

Our Conception of Dialog Support 1a

RECORDED DIALOG 1a1

One of the prime objectives of the augmentation system developed at ARC is to aid collaborating knowledge workers by providing flexible computer tools and methodology for communicating with one another. We collectively refer to such tools and methodology as a Dialog Support System (DSS). Its primary task is to provide mechanisms for transmitting on-line messages and documents between users. However, for large projects or those about which some larger community of users must remain informed, the dialog soon becomes unmanageable without additional computer aids. ARC's DSS therefore: 1a1a

1) permanently records (copies to read-only storage),
2) numbers (assigns a unique accession, or catalog number),
3) and catalogs (records author, title, number, and location) 1a1a1

each piece of dialog -- for later consultation, for reference by later documents, and for examination by interested bystanders. 1a1b

THE JOURNAL 1a2

ARC's DSS is implemented as a set of computer processes called the Journal, consisting of a foreground subsystem which interacts with the user and provides primitives for entering a message or document in the Journal (with title, author and other information), reserving catalog numbers, and so forth, and a background process that further processes submission requests and delivers mail to the addressees indicated by the author. The Journal is supported by several additional systems: an Identification System responsible for maintaining information about users -- their location, group memberships, phone numbers, and so forth -- and a Number System responsible for keeping track of which catalog numbers have been assigned and to whom, and which are available for future assignment. 1a2a

Since its implementation in April of 1971, the Journal has been heavily used (containing at present over 10,000 messages and documents), initially by the ARC staff, then by a larger user community with network access to ARC's

computer facility, and most recently by commercial and government users of a second computer facility operated for ARC. The Journal has evolved as a result of our experience with it and in response to the increased demands placed upon it by its growing user base. This report describes that experience and evolution. 1a2b

Our Initial Implementation 1b

THE ARC/NLS ENVIRONMENT 1b1

ARC's DSS resides on a heavily loaded Digital Equipment Corporation (DEC) PDP-10 running Bolt, Beranak, and Newman's (BBN) Tenex operating system. Tenex provides a time-sharing environment in which 10-20 users independently interact with any of a variety of applications packages called "subsystems". ARC's PDP-10 is devoted almost exclusively to providing access to a single subsystem, ARC's On-line System (NLS) (13041,), a comprehensive system of tools for manipulating structured text. 1b1a

NLS provides a very general set of primitives for manipulating and viewing tree-structured text files. Commands are provided for manipulating the tree's structure, e.g., for adding nodes called "statements" to the tree, for deleting single statements or whole branches of the tree, for moving or copying a subset of the tree from one location to another, and so forth. 1b1b

In order to maintain flexibility in the first implementation and to facilitate maintainence of the system, NLS text files were consistently used in implementing the Journal, Identification, and Number Systems' principal data bases, as well as for catalogs, indices, and a variety of internal, inter-process communication files. 1b1c

STRUCTURE 1b2

The Journal 1b2a

The Journal System is a set of procedures that run in both foreground and background modes to maintain a data base of recorded documents, and to distribute them to specified addressees. 1b2a1

Larger Journal documents are stored as separate files in a set of system directories. Short documents, called "messages", given special treatment in the interests of economical storage, are stored in a set of (currently

about 20) files, several hundred to a file. Whenever a document remains unreferenced for a month, it is archived to magnetic tape by Tenex, and its on-line storage released for other use. Although over 10,000 items have been journalized on the PDP-10 since April of 1971, most have long ago been archived and therefore do not occupy on-line storage, except when brought back for reexamination. 1b2a2

The Journal maintains a system catalog of all recorded documents, implemented as a set of (currently five) on-line files. The catalog contains information used by NLS to locate a Journal item given its catalog number, as well as information used by stand-alone programs to produce non-system catalogs and indices (by author, titleword, and number). 1b2a3

Journal mail addressed to a particular user is delivered in one or both of two delivery modes, On-Line and Hardcopy. The delivery parameters are selected by the addressee and maintained by the Identification System. A document's author need know nothing about the delivery mode of its addressees. 1b2a4

ON-LINE DELIVERY 1b2a4a

Regular users of NLS normally receive on-line delivery of all their Journal mail. Each item is placed by the Journal in a special NLS file called the user's "initial" file (so named because the file's name is the user's ident, which in turn is usually his initials). For convenience, this file is automatically loaded for the user when he enters NLS. The text of short messages is delivered to the user in its entirety. For longer items, only a citation giving the document's author, title, and date, and a convenient, machine-readable pointer (called a "link") to the text of the document are delivered. 1b2a4a1

HARDCOPY DELIVERY 1b2a4b

Hardcopy line printer output is sent by U.S. Mail to users who never or only infrequently use NLS or who, for one reason or another, want it in place of or in addition to on-line delvery. A substantial amount of clerical support is required to support hardcopy delivery. 1b2a4b1

The Journal maintains information about on-going distribution operations in a single, NLS file, used also as a vehicle for communication between the submission and distribution components of the background system. 1b2a5

The Identification System 1b2b

The Identification System is a set of procedures that maintain a large data base, implemented as a single, very large NLS file, containing information about individuals, groups of individuals, and organizations (each of which is assigned a unique name called an "ident"). Various information fields are maintained for each ident, and procedures are provided for manipulating each field. 1b2b1

The Identification System includes an NLS subsystem that permits users to interrogate and modify the data base themselves, subject to the appropriate access controls. 1b2b2

Because of the data base's size, and because updating the data base involves creation of a new version of the file (requiring about 30 seconds or more of real time on a loaded system), all of the changes for a particular ident are collected from the user before the file is updated. 1b2b2a

The Number System 1b2c

The Number System is a set of procedures that manage a data base, implemented essentially as a single NLS file, containing information about the assignment of catalog numbers to Journal documents. The data base contains 1b2c1

1) a number of blocks of numbers available for assignment. 1b2c1a

2) a list of assigned numbers (either recently used, assigned but as yet unused, or in the process of being used) and for each the date and time of assignment and the ident(s) of the user(s) to whom it was assigned. 1b2c1b

It is often useful to know in advance what number will be assigned by the system to a particular item. This is necessary, for example, to create a set of documents that internally reference one another. A catalog number may therefore be reserved for later submission, or "preassigned". 1b2c2

The RFC number system, a separate special-purpose number

system patterned after the master system (and thus able to use most of the same primitives), was implemented at the request of an informal group of network protocol developers. An item may have an RFC number in addition to the master catalog number. 1b2c3

EXPERIENCE AND PROBLEMS 1b3

A number of problems with the initial Journal implementation have been encountered and attacked. Some of the major problems are described below. 1b3a

Excessive real-time required for submission 1b3a1

In the initial implementation, the entire submission process, with the exception of delivery, was performed in the foreground and therefore kept the user from other work for what often, given the system load, proved to be an inordinate amount of time. In an attempt to alleviate this problem, the submission mechanism was restructured, and all manipulation of catalog, distribution, and storage files deferred to the background process. 1b3a1a

A special system directory was established for queuing submission requests for the background process, that now goes through two distinct phases. First, all queued submissions are processed: numbers are assigned where necessary, the document is stored in the appropriate message or separate file in the appropriate system directory, the document is cataloged, and a distribution request is queued. And second, whatever distribution requests have accumulated are processed, one individual addressee at a time. 1b3a1b

To further reduce the amount of processing that must take place in the foreground, a form of submission is permitted in which the task of assigning a catalog number is deferred to the background process. Deferred submission is the default, and most submissions are therefore of this type. Since deferred submission does not require write access to any system files, a user can submit an item in this mode at any time, regardless of the state of the Journal or Number System files. 1b3a1c

Background delivery degraded system performance 1b3a2

The Journal background process has proven to be very expensive to run, and often has had a detrimental effect upon the responsiveness of the system as viewed by its interactive users. We've experimentally varied the frequency with which the background process runs (and thus with which mail is delivered), from once per day initially, to its current frequency of once every hour. 1b3a2a

The background process now periodically checks the load average (the Tenex monitor's measure of system demand) and suspends processing if it is above some predetermined cut-off value. Processing is resumed only when the load average drops sufficiently. The check is performed at a point in the process when the system files are consistent and least vulnerable to a crash. Between these check points, the process runs at high priority. 1b3a2b

The benefits of this strategy are threefold: the background process does not add appreciably to the system load when it's already high; it can exploit slack times throughout the day; and since the probability of a crash increases with system load, the Journal and Number System files are usually in a relatively invulnerable state when a crash occurs. 1b3a2c

Data bases vulnerable to system failures 1b3a3

A very serious problem of the initial Journal implementation was the vulnerability of the various system files to hardware (especially disk) problems, monitor crashes, and exhausted disk storage. The processing of hardcopy output, besides being time consuming, was similarly vulnerable to both software and hardware failures. 1b3a3a

The danger of losing system files because of lack of disk storage has been greatly reduced by also checking for available disk space at the same time the load average is checked. Processing is terminated until the next hour if space is too low. This strategy prevents loosing a system file due to exhausted disk space during a file update. 1b3a3b

A number of problems associated with the processing of hardcopy output have been largely eliminated. A variety of monitor bugs have been fixed or avoided. The bulk of the processing is done during the evening

or early morning hours. Because of the volume of hardcopy output produced by the Journal, the print requests were first placed on magnetic tape and printed on an IBM 360 system elsewhere at SRI, and finally contracted outside of SRI. Network delivery, described in the next section, has, on the other hand, drastically reduced the volume of hardcopy produced, and thus recently permitted us to resume printing on our own system at OFFICE-1. 1b3a3c

Extensions for a Network Environment 1c

THE ARPANET ENVIRONMENT 1c1

In July of 1970, ARC's PDP-10 became part of the ARPANET, now an international network of large-scale computer facilities called "hosts" linked by 50 kb communication lines. Once the lowest level, inter-machine communication protocol was developed, the central task was to design and implement the software protocols required for general, inter-process communication and other, more specialized exchanges. This task was undertaken by an informal group of geographically separated systems programmers called the Network Working Group (NWG). 1c1a

In early 1969, ARC had offered to serve as the Network Information Center. As soon as hardware connections were made and protocol development reached a stage sufficient to permit simple, teletype-like use of a remote time-sharing system, ARC began to provide dialog support for the NWG via the Journal. 1c1b

JOURNAL CHANGES TO SUPPORT THE NETWORK 1c2

At first, the Network user used the Journal in nearly the same manner as a local user. Like local users, he had to login to the ARC system and use NLS to compose and journalize a document. But unlike most local users, he received hardcopy, rather than on-line delivery of his Journal mail. When ARPANET protocols developed to the point of permitting the transmission of text files and mail to users at remote hosts via the Network itself, the Journal was modified to utilize this new capability. 1c2a

Network delivery 1c2b

The File Transfer Protocol (FTP) (17759,) devised by the NWG permits the transmission of text to a named "mailbox" at a remote host. For purposes of receiving mail,

therefore, each Network user has a "network address" consisting of a host name and a mailbox name. To exploit this new Network capability, we added a third, "network" delivery mode to the existing on-line and hardcopy modes, storing a network address in the ident file for each Network user. A Network user can thus take delivery of all Journal mail addressed to him, in his own system, simply by storing the appropriate delivery parameters in the Identification System. 1c2b1

Rather than deliver extremely long documents via the Network in their entirety, we made the same size distinction for network delivery as for on-line delivery, sending only a citation for long documents. We modified the FTP software supplied by BBN to recognize a distinctive pathname (that the Journal provides with the delivered citation) that, when used to retrieve Journal documents, invokes a conversion of the tree-structured document to sequential form before transmission through the Network. A Network user can thus explicitly retrieve the full text of any Journal document sent to him. 1c2b2

Network submission 1c2c

The fact that the Network user had to explicitly connect to and login at ARC's PDP-10 to enter a document into the Journal, and that he had to compose the document using NLS, complicated life for some users, forcing them to learn the details of NLS, in which some had only one, specialized interest. 1c2c1

To alleviate this problem, we implemented a facility that permits users to journalize documents composed via their local editor without explicitly connecting to the ARC system or logging in, and without any knowledge of the NLS command language. We did this by further modifying BBN's FTP software to recognize a special mailbox name of the form "authors/addressees" and to interpret it, in the context of a mail delivery, as a Journal submission. The ident lists "authors" and "addressees" are verified by NLS, running beneath the FTP program in an inferior fork. If the ident lists are found correct, the "mail" is immediately journalized. Thus the remote user can journalize a document using the normal, Network mail facility provided by his system. 1c2c2

EXPERIENCE AND PROBLEMS 1c3

The Journal's Network submission and delivery facilities

have been in operation for over a year now. The latter has suffered from a few, relatively minor problems. Network addresses, for example, are not well understood by some users who, in attempting to modify them themselves, have frequently modified them incorrectly. In such cases, delivery of the user's mail is prevented until the error is discovered and corrected by ARC personnel. Because of this, almost all Identification changes are now done by ARC staff. Many users are unwilling to explicitly retrieve the text of long documents for which they are sent only a citation, even though the retrieval process is straightforward, even automatable. 1c3a

The submission facility suffers from more severe problems, one of which is that the ident verification and journalization processes are very time consuming and must be completed before the user's request is acknowledged and he is "set free". A more satisfactory strategy would be to queue the request and acknowledge it immediately, releasing the user for other work, and then to perform the expensive processes in background mode, with a Network message sent to the author in case of failure. 1c3b

A second problem is that the conversion that the Journal must make between the sequential text file presented by the user and the tree-structured NLS file required by the Journal is often unsatisfactory to the user. We believe this to be a very difficult problem to solve, one perhaps best handled by permitting the inclusion of sequential files in the Journal data base, thereby eliminating the need for conversion. 1c3c

A final problem is the inadequacy of the mail subset of the FTP protocol, which makes it difficult or impossible for the user to transmit any of the optional parameters supported by the Journal, and which forces the user interface to remain somewhat artificial. ARC has proposed a separate mail protocol (17140,), but no protocol development is being carried out in that area at present. 1c3d

Extension to a Dual-Site System 1d

THE SRI-ARC/UTILITY ENVIRONMENT 1d1

In January of 1974, ARC began operation of a second, "Utility" PDP-10 system we call OFFICE-1 to provide NLS support in a stable environment to what has proved to be an ever-growing clientele. The facility is operated for ARC by Tymshare, Inc. from Cupertino, California. Like ARC's own

PDP-10, OFFICE-1 is connected to the ARPANET, through which most of its users gain access to it. The Utility's software configuration is essentially identical to ARC's, providing the full range of NLS service to its users. One such service is, of course, the DSS. 1d1a

In providing Journal service from the Utility, we decided to include that second system within the domain of what is conceptually a single Journal spanning both the ARC and Utility machines. That is, rather than simply replicate the software, thereby creating a second, independent system, we decided to couple the two DSS systems, making all items journalized from either system available at both and addressable to users resident on either machine. Thus, for example, we employ a single Ident File, but maintain it in duplicate. 1d1b

STRUCTURAL CHANGES 1d2

In implementing a dual-host Journal, we were somewhat pressed for time and therefore decided to design and implement a interim system and later replace it with a more efficient and carefully thought out implementation. 1d2a

The interim dual-host Journal we decided upon involves duplicate Journal, Identification, and Number Systems, cognizant of each other at only a few points in the code. The two systems communicate with one another through the ARPANET via FTP. We implemented a special, assembly-language module to perform the FTP operations on NLS's behalf, since the corresponding FTP software provided by BBN is neither designed to be called by another program (since it's implemented as an interactive subsystem) nor structured in such a way that the relevant subroutines can be easily extracted. The portion of BBN's FTP software that was retained has been modified to deal more satisfactorily with NLS files, which have blank spots in their address space. 1d2b

Two Journal Systems 1d2c

Each submission request, regardless of its source, is fully processed by the Journal System on each machine. Each system's Journal catalog and document files, though in a sense maintained independently, are always identical (neglecting the obvious time lag). To avoid duplicate delivery of each Journal item, as would naturally occur as a consequence of duplicating the submission request, we partitioned the idents, assigning responsibility for

delivering mail to any particular user to (in most cases) just one of the two systems -- the one on which the user does most of his work. 1d2c1

Submission requests are duplicated in the following manner. The background process on each system, before processing recent submissions, moves any files in the other host's special communication directory (OUTJOURNAL) to a local submission queue directory (TEJOURNAL), thus adding them to the list of local submissions to be processed. Then, in processing that list, a copy of each submission request, except those obtained from the second host, is queued for the other system in the local communication directory (OUTJOURNAL again). 1d2c2

Two Identification Systems 1d2d

To simplify the task of uniting the two Identification Systems, we bypassed the problem entirely by permitting additions and modifications from only one machine. The other machine is periodically sent an updated copy of the entire data base. 1d2d1

Two Number Systems 1d2e

The two Number Systems function independently, each assigning catalog numbers from a separate block. Numbers preassigned on one machine must be used on that machine, and the RFC Number system is available on only one machine. 1d2e1

EXPERIENCE AND PROBLEMS 1d3

Aside from the obvious inefficiency of duplicating each submission on the remote machine even though the item may be of only local interest, there have been no serious problems with our interim implementation. 1d3a

An occasional asynchrony problem arises as a result of the time delay between an addition or modification to the ident file and receipt of the modified version of the data base at the second machine. For instance, an ident could be added to the Identification System, a Journal item sent to him from that machine (which already knows of his existence), and the item could reach the remote system via FTP before that system becomes aware of his addition to the system, causing an error in the remote system's Journal delivery function. 1d3b

The most common problem with the dual-host system is Network transmission errors during file transfers. Such failures cause the item being transmitted to be delayed until an operator finds the file in an unusual state on the source machine. He must then check the destination system to verify that the file has not in fact arrived (which is the usual case), and then requeue it for transmission. Since occasional Network failures are inevitable, we are attempting to enhance the performance of the dual-host system by automating the detection and requeuing process. 1d3c

The redundancy of information within the dual-host system is occasionally useful for reconstructing data lost due to a malfunction of the file system. A backup of the file system recently experienced by the Utility cost no more than reconstruction time; no Journal files were lost. 1d3d

Private Dialog 1e

COMING TO GRIPS WITH THE PROBLEM 1e1

From the outset, one of the design goals for the Journal has been to provide an atmosphere in which memos, formal design documents, proposals, and other items, once published, would thereafter be readily accessible to anyone who cared to consult them. Author and subject indices are periodically produced and anyone, whether an active participant in the dialog or not, can therefore browse through the list of items authored by a particlar individual or written on a particular subject, skimming or reading in full any items that look useful or appealing to him. 1e1a

This model of dialog was appropriate for the system's initial user community, ARC itself, where subgroups working on highly inter-related tasks must keep abreast of one another's activity. As the Journal's user community grew to encompass researchers throughout the ARPANET, the model remained for the most part appropriate. Again the participants were engaged in separate but inter-related subtasks of a single, large project (i.e., ARPANET protocol design and implementation), and each working group had legitimate (and often vital) interest in the work of the others. But with the extension of the Journal to a dual-host system, a new class of users became involved. Many Utility users, though anxious to use the Journal as a dialog support aid, were not at all anxious to have all of their dialog (including, perhaps, personal correspondence, new product information, and so forth) accessible to the general public. Thus ARC was compelled to address itself to

the problems of non-public, or private dialog, and to provide support for it through the Journal. 1e1b

CHANGES TO THE JOURNAL 1e2

What follows is a brief discussion of the more fundamental implementation problems which we encountered in tackling this problem; the reader is referred to (22911,) for a more detailed statement of the Journal changes made. 1e2a

The Journal must ask itself three questions in establishing a user's right to view a recorded document: 1e2b

1) Who is it requesting access to the document? 1e2b1

2) Has he explicitly been granted access to the document? 1e2b2

3) Is he a member of any group (perhaps by way of one of more levels of indirection) that has been granted access to the document? 1e2b3

Who is the requestor? 1e2c

The Journal has always tolerated imposters, simply accepting the user's word for the ident he declares at login to be his. It has done so because it could afford to, and because it was difficult to do otherwise. 1e2c1

Access to a user's personal files is controlled by the monitor, and all system files (i.e, Journal documents) were accessible to everyone. The only thing which hinged on the ident claimed by the user was the authorship of items he journalized during the session. 1e2c1a

Since the Journal designates users by ident, rather than by directory name, and since elements of the two name spaces cannot, in general, be placed in one-to-one correspondence (several users, each with an ident, often sharing a single directory), the monitor's login identity check was of little use as it stood. 1e2c1b

Rather than significantly perturb the Tenex login procedure, we adopted the following strategy. 1e2c2

For those users who have personal directories, we constructed a system data base giving ident as a function of directory. Tenex was modified to infer the user's ident from his stated directory name

(which, of course, had to be accompanied by the appropriate password) at login, using the data base, and to store it in a read-only, job-global cell for subsequent interrogation by NLS. 1e2c2a

For those users who share a directory, we placed opposite the directory name in the data base the idents of the users who use the directory. When Tenex encounters such a user at login, it interrogates him for his ident, accepting only one that appears in the list. 1e2c2b

Thus, those users who are assigned a personal directory, and who login only under that directory, are completely protected by the system (i.e., they cannot be impersonated), while those who work in a community directory, are less fully protected, since they can be impersonated by any other member of the directory community. We are encouraging user organizations to set up separate directories for each user. 1e2c3

Has the requestor been granted access to the document? 1e2d

We have defined two classes of Journal items: private and public. Whenever a document is entered in the Journal, its author can select the class most appropriate, with public being the default. Private documents are defined to be readable only by the clerk, an author, or a distributee. That list of idents, including in general those both of individuals and groups, is stored as text in the first statement of the file which ultimately holds the document in read-only storage. Whenever a user attempts to load the file, the list is consulted, and if the requestor's ident appears in it, his request for the document is honored. 1e2d1

Has he been granted access by implication? 1e2e

Since authors and distributees may be groups of people (or other groups), as well as individuals, the access list for a private document in general contains group, as well as individual idents. A user who requests access to a private document may therefore have legitimate access to it by virtue of his membership in a group, without his individual ident appearing explicitly in the access list. Because group idents are used heaviliy is this way, we were compelled to provide efficient means for verifying an ident's IMPLICIT appearance in an access list. 1e2e1

To this end, the Identification System was modified to maintain back links, as well as forward links between each group ident and the idents of its members. That is, not only is a membership list maintained for each group ident, but in addition, now, a group list is maintained for each individual or group ident, specifying the list of groups of which the ident is a member. 1e2e2

The logged-in user's group list is loaded by NLS once per session, and by a simple search of that list, most instances of legitimate access attempts to private documents can be identified. For those cases in which the user's claim to a document is more complicated (e.g., requestor A is a member of group B that is a member of group C, that appears in the access list), the Identification System is consulted and its data base examined more thoroughly. 1e2e3

EXPERIENCE AND PROBLEMS 1e3

The private dialog feature of the Journal has been in advertised use for only a few months, and hence any in depth attempt to evaluate its performance or use would be premature. The areas in which effects are most likely to be expected are those involving intimate collaboration between users. It's long been common practice, for example, for cooperating users to impersonate one another to get at a file which, though necessarly residing in one particular directory, is in reality a joint file. In implementing private dialog, we've necessarily restricted such practices, and the result will probably be the design and implementation of more formal methods for accomplishing such shared tasks. 1e3a

Our Thinking About a General, Multi-Site System 1f

MOTIVATION 1f1

Recognizing the immediate need to provide dialog support for Utility users, and recognizing also that the implementation of an efficient dual-host dialog support system would require significantly more than simple modification of the existing, single-host system, we elected to make the short-term modifications described earlier and then to begin design work on a general, multi-host system to be distributed on an arbitrary number of ARPANET host systems. 1f1a

The implementation of such a system would involve a complete rewriting of the present Journal, Number, and Identification

Systems. Furthermore, we expect that the new DSS will in many ways be a different system, one in which many of the basic concepts of the previous system find a place, but also one in which new concepts appear. 1f1b

DESIGN GOALS 1f2

In designing a Multi-Host Journal System (MHJS), we had a number of goals in mind, the first necessarily being modularity: 1f2a

Modularity 1f2a1

We envision a system composed of modules, each providing some specialized service to the others, or to the end user, and which together comprise a coherent system. 1f2a1a

Each module implements a set of primitives whose syntax and basic function are to be standardized, but whose internal workings would be left unspecified by the design (within certain broad constraints), being dependent upon the implementation machine, and the particular role that the module is to play within the System as a whole. 1f2a1b

Reconfigurability 1f2a2

The MHJS must be reconfigurable. Although the design suggests in broad terms the manner in which the System is to be constructed from its component modules, the design does no more than specify a family of MHJS's from which a particular configuration can be selected (in the same way that a computer system manufacturer provides a set of hardware modules (disk drives, CPUs, etc.) from which the customer configures his particular system). 1f2a2a

The design specifies a small set of module types, each of which is replicated in appropriate numbers for a particular system configuration. 1f2a2b

The MHJS must be reconfigured, for example, to accommodate the addition of new hosts to the system, or it might be reconfigured to place an instance of a frequently used module closer to a population center, or for any of a variety of other reasons. 1f2a2c

Optimum Data Base Distribution 1f2a3

It is, of course, more expensive to manipulate remote data bases than local ones; sometimes it's impossible (e.g., when the remote host is down). The MHJS, therefore, must attempt to reduce the frequency with which remote data bases must be dealt with by replicating portions of them in centers of user population and message traffic. 1f2a3a

Uniform and Consistently-Applied Access Controls 1f2a4

The MHJS must recognize the existence of private information of every type (documents, catalogs, idents, etc.) and provide the access controls necessary to protect it, providing for private dialog of a much more flexible nature than that described in the preceeding section. 1f2a4a

With these goals in mind, then, we began designing a Multi-Host Journal System. Some of the more important concepts we came up with are described below; the reader is referred to (23144,) for a more complete discussion. 1f2b

SOME IMPORTANT CONCEPTS 1f3

Isolating the Recording, Cataloging, and Distribution Functions 1f3a

The original Journal implemented a single user primitive we called "submit" which records, catalogs, and distributes a document. We considered that primitive fundamental to dialog support, and the vision of it colored our thinking about the Journal's internal structure. We've since learned that the sub-primitives from which submit is constructed are also of interest to the user. 1f3a1

For example, we've found it useful to be able to distribute a previously submitted document to additional users, an operation that we've implemented and call "secondary distribution" (even the name reflecting our bias toward "submit"). We now recognize, further, the need to be able to distribute a document without recording it at all, a facility that the present Journal still does not offer. And we recognize the cataloging sub-function of "submit" to be a more generally useful tool, applicable, for example, to personal as well as system data bases. 1f3a1a

Access Controls 1f3b

We decided from the outset of the design to implement flexible access controls throughout the MHJS, applying them not only to documents, but to data elements of all types -- catalogs, idents, and so forth. Controlling access to a data element consists of specifying, when the data element is created, the list of individual and/or group idents granted access to it, and then limiting access to members of that list. 1f3b1

This is the same kind of access control now implemented in the present Journal, as we've already described, and is by far the most satisfactory type of which we know. In the MHJS, we've taken the additional (and natural) step of assigning passwords to idents, and requiring their use, as a means of verifying the user's identity. 1f3b2

Catalog Number Assignment 1f3c

The present Journal assigns every recorded document a unique identifier called a catalog number by which the document can be referenced or retrieved. Since the MHJS is conceptually a single Journal, we must somehow maintain uniqueness in catalog number assignment, while yet hopefully making the assignment process reasonably efficient and reasonably insensitive to host failures. These requirements preclude the simplest implementation, i.e. assignment of numbers by a single module at a single host. 1f3c1

The approach we think most satisfactory is to station several instances of a module we've called the Number Vendor at strategic points about the system. Each additional Number Vendor, assuming it resides on a different host, increases the probability of a user's being able to obtain a catalog number when he wants it, as well as reducing the overhead (by placing the source closer to him). 1f3c2

At any point in time, each Number Vendor owns a subset of the universe of catalog numbers from which it can satisfy user requests. A Number Vendor may only assign catalog numbers that it itself has been assigned by another Number Vendor, except for one special "root" number vendor assigned initial possession of the entire name space. 1f3c3

Number Vendors might be stationed throughout the MHJS, each with responsibility for servicing a segment of the user population, and each replenishing its number supply,

when it nears bottom, from the root vendor. This strategy permits a form of number assignment that is both efficient and insensitive to the host failures that periodically make the root Number Vendor inaccessible. 1f3c4

Publishing a Document 1f3d

In our design of a MHJS, we've made central a concept that is only given lip service in the present Journal, that of subcollections. A subcollection is a subset of all recorded documents, each of whose members shares some common attribute, e.g. author, subject, and so forth. A single document may be assigned to zero or more subcollections, either explicitly by the author, or implicity by the system. Although hardcopy subcollection catalogs can be generated, the Journal maintains no on-line subcollection catalogs, thus severely limiting the utility of the concept in its present implementation. 1f3d1

A major concern of the MHJS is to provide specialized marketplaces in which documents can be exchanged. Such a marketplace is called a "forum", and one speaks of "publishing" a document in a forum. In the MHJS we've thus placed great stress on the concept of allying a recorded document with other documents related to it (i.e., placing it in a subcollection), relegating the concept of simply recording a document to a less central role. 1f3d2

Users with interest in a particular forum can formally declare that interest, and, subject to appropriate access controls and accounting disciplines, become "subscribers" of it, thereafter automatically receiving an announcement of each new document published. The prime responsibility of the Publisher, the module which implements a forum, is therefore to catalog each document as it is contributed, and send a copy of the catalog entry (giving the document's author, title, date of publication, etc.) to each of its subscribers. We've thus given the old concept of subcollections an active, rather than passive character, with the system notifying interested users as new documents are made available. 1f3d3

Maintaining Networks of Documents 1f3e

For reasons of efficiency and reliability, it is necessary to permit an arbitrary number of physical copies of a document to exist simultaneously within the MHJS. Each additional copy, assuming it is created on a

different host, increases the probability of a user's being able to retrieve the document when he wants it. A retrieval request can be satisfied most quickly, of course, if a copy of the requested document happens to exist on the user's own host already. The system might therefore create a copy of the document at each major population center, anticipating a rash of retrieval requests; and then delete the copies a month later, once the period of peak demand has passed. 1f3e1

Access to a document and all its copies is uniformly controlled on the basis of access lists assigned by the author. A user, for example, cannot read a document unless the author granted him read access to it. The copying of documents, however, is a system function designed to promote efficiency and is therefore unhindered by access controls. 1f3e2

Each recorded document within the MHJS is therefore implemented as a network of copies whose topology is a dynamic characteristic of the system and changes with such things as the frequency with which it is referenced. The system keeps track of the various copies of a document, and can thus direct the curious user to the nearest one. 1f3e3

Distributing Information About Users and Modules 1f3f

A need that pervades the MHJS, even more so than in the present Journal, is that of swift access to information about users of the system. In the present system the data base is called the Ident File and describes the users and user groups known to the system. To implement the access controls that the MHJS seeks to maintain throughout, both human users and system modules are assigned idents. Group idents are very heavily used, being extremely convenient for implementing access lists for the various data bases within the system. 1f3f1

For reasons of efficiency and reliability, it is highly desirable to maintain copies of subsets of the Ident File at various locations within the system, each under the control of a module called a Registrar. An ident can be known to an arbitrary number of Registrars, and that particular set of Registrars is called the ident's "domain". Information about the ident can be obtained from any Registrar in its domain. Modifications to an ident are relayed to all Registrars affected, by the Registrar that receives the modification request. 1f3f2

The Registrar turns out to be the workhorse of the MHJS, and its importance cannot be underestimated. In designing the MHJS we discovered that: 1f3f3

1) Virtually every system module must deal with incidental data bases which are lists of user/program names (e.g., access lists), and each must provide mechanisms for retrieving and modifying them. 1f3f3a

2) System modules can be relieved of a significant burden by providing a specialized module (the Registrar) whose function is to provide the primitives required to manipulate these data bases. 1f3f3b

3) Furthermore, the lists then become accessible from any one of an arbitrarily large set of Registrars (the group ident's domain), since the Registrar already implements the required broadcast facility. 1f3f3c

4) Since the existence of a document's read access list (for example) implies the existence of the document itself, whether or not a document exists can be determined by consulting the nearest Registrar. 1f3f3d

5) Race conditions associated with the creation of a document (e.g., two users attempting to create a document with the same catalog number simultaneously at two different points in the system), for example, can be arbitrated by the use of locking mechanisms implemented by the Registrars. 1f3f3e

Conclusion 1g

Having made heavy and continuous use of the Journal for over three years now, ARC has found it to be a powerful dialog support tool for knowledge workers. 1g1

During the course of its use, the Journal has been substantially modified to increase its efficiency, extend its geographical reach, and provide the new features we've discovered to be important. Initially an experimental system supporting a fairly small number of geographically concentrated researchers, it now supports a large, geographically distributed user community linked by the ARPANET. Initially a software system implemented on a single computer, it now operates on a pair of PDP-10 systems linked by the Network, and design work has been done for a general, multi-host system. Initially exclusively a forum for public dialog, it now supports private communication as well. 1g2

The Journal will further evolve and new features will be implemented and experimented with as we continue to gain experience in the dialog support field. 1g3

ARC Dialog support: ROUGH DRAFT for a Report Chapter

(J24450) 8-NOV-74 15:01;;;; Title: Author(s): James E. (Jim) White/JEW; Distribution: /DVN([INFO-ONLY]) RWW([INFO-ONLY]) ; Sub-Collections: SRI-ARC; Clerk: DVN;

The Dictionary Has Two p's.

See (vannouhuys,novguide,"equipped") 1

The Dictionary Has Two p's,

(J24451) 8-NOV-74 15:20;;;; Title: Author(s): Dirk H. Van Nouhuys/DVN; Distribution: /RWW([INFO-ONLY]) ; Sub-Collections: SRI-ARC; Clerk: DVN;