KIRK 26-FEB-75 20:09 25500

Alternating ODD,EVEN in footers

```
@F='@Font=6ps NLS-8 @Splits @Font=10p,5$ @GPN$' , '@Font=10p,5$ @GPN$
@Splits @Font=6p,6$ GLOSSARY'$
```

KIRK 26-FEB-75 20:09 25500

Alternating ODD,EVEN in footers

This doesn't work to set odd and even page number formats. What I Get instead is
NLS-8 1" , " 1 GLOSSARY
at the bottom of every page (1 increments to contain proper page number). What am I doing wrong. I think it is exactly like the latest documentation says. 1

KIRK 26-FEB-75 20:09 25500

Alternating ODD,EVEN in footers

(J25500) 26-FEB-75 20:09;;;; Title: Author(s): Kirk E. Kelley/KIRK;
Distribution: /NDM([ACTION]) EKM([INFO-ONLY]) DVN([INFO-ONLY]
) ; Sub-Collections: SRI-ARC; Clerk: KIRK;

SGR 26-FEB-75 22:33 25501

No More Identfile Changes

The identfile has been frozen (with ARC's new quick freeze method) to fix all delivery info before the move this weekend. Any changes made between now and Monday will not be carried over. 1

SGR 26-FEB-75 22:33 25501

No More Identfile Changes

(J25501) 26-FEB-75 22:33;;;; Title: Author(s): Susan Gail Roetter/SGR; Distribution: /SRI-ARC([INFO-ONLY]) ; Sub-Collections: SRI-ARC; Clerk: SGR;

SGR 27-FEB-75 09:20 25502

User Services March Training Schedule

The following is a fairly firm schedule of training for the month of March. Rita will probably spend the first two weeks in Washington helping Jeanne and the last two weeks traveling and helping me with training. 1

3-5 6-7 10-12 13-14 17-18 19-21 24-28 2

SGR ETS NEW LONDON CONN MCA PANAMA CITY FL SAN DIEGO ARC ARC 3

New London, Conn, Panama City, Florida, and San Diego are sites planning to do collaborative work with NSRDC using their two slots. 3a

3-4 5-6 7 10 11 12-14 17-21 24-28 4

JMB ARPA NSRDC ARPA ARPA ARPA-NSW ARPA GUNTER & ARPA ARPA + OTHER 5

ARPA-NSW refers to the NSW people in the Pentagon. 5a

SGR 27-FEB-75 09:20 25502

User Services March Training Schedule

(J25502) 27-FEB-75 09:20;;;; Title: Author(s): Susan Gail Roetter/SGR; Distribution: /SRI-ARC([INFO-ONLY]) ; Sub-Collections: SRI-ARC; Clerk: SGR; Origin: < ROETTER, SCHEDULE.NLS;1, >, 27-FEB-75 08:49 SGR ;;;;####;

DVN 27-FEB-75 10:27 25503

XGP Commands Branch Running at ARPA

I'm glad to hear it's doing some good for you. When the simpler TENEX command is available, Jeannie sould try to wrap the whole thing into one package. If she needs help I will be glad to do what I can. 1

DVN 27-FEB-75 10:27 25503

XGP Commands Branch Running at ARPA

(J25503) 27-FEB-75 10:27;;;; Title: Author(s): Dirk H. Van Nouhuys/DVN; Distribution: /JOAN([ACTION] dpcs notebook please) CKM([ACTION]) JMB([ACTION]) RWW([INFO-ONLY]) DCE([INFO-ONLY]) ; Sub-Collections: DPCS SRI-ARC; Clerk: DVN;

CHI 27-FEB-75 12:16 25504

The Control Meta Language -- A Formal Language for Describing and Implementing Interactive Control Languages

submitted for 1974 final report chapter.

Title: The Control Meta Language -- A Formal Language for Describing and Implementing Interactive Control Languages 1

Authors: Charles H. Irby, Charles F. Dornbush, Donald C. Wallace 2

Augmentation Research Center, Stanford Research Institute 2a

CONTROL META LANGUAGE -- CML 3

INTRODUCTION AND MOTIVATION 3a

To facilitate the easy formal description, implementation, and modification of the user interface to a range of interactive application programs, the Augmentation Research Center (ARC) has developed the Control Meta Language (or CML). This was an outgrowth of earlier efforts to accomplish the same goals at ARC [1,2,3,4,5]. 3a1

The goals of this development were the following: 3a2

1) Provide a means for easily changing and experimenting with the user interface to an interactive application program. 3a2a

2) Allow for the independent manipulation of 3a2b

a) the commands available to the user and 3a2b1

b) the interaction methodology and techniques that are used to specify commands. 3a2b2

3) Provide builders of new interactive application programs with a facility for easily creating the user interfaces for their new programs. 3a2c

4) Provide the user with consistent and coherant command language features across a collection of application programs, or what might be termed "tools". 3a2d

Independent of the tool to which the user is giving commands, he does so using the same methods for specifying which commands he wishes executed, the same methods for specifying arguments or parameters to commands, gets the same type of prompting and requests help in the same way, always. In addition, the general syntactic form(s) of commands should be the same from tool to tool unless there is good reason for the tool to deviate from the standard. Of course the particular

commands and vocabularies will vary with the tool and in fact the same verbs may be used with quite different semantics in different tools, but at least most other facits of the command language (including asking for help and being prompted for the proper type of input) should stay the same across tool boundaries. 3a2d1

2) To provide tools with well-formed commands. 3a2e

Many operating systems and application programs have elected to use half duplex, line-at-a-time terminals because of the increased computer efficiency provided by this approach. Other operating systems and application programs have chosen, instead, to utilize character-at-a-time full duplex terminal disciplines because of the opportunity this provides for utilizing a more human-engineered command language. 3a2e1

The CML system is an attempt to combine these two approaches into a COMMAND-AT-A-TIME system, where the application programs do not directly interact with the terminal, but rather receive fully specified commands from the Frontend. At the same time, the CML interpreter will attempt to provide the user with the best possible human-engineered command language discipline. 3a2e2

Although initially this was done by issuing direct procedure calls on tools (requiring that the CML interpreter and tools be written in the same language and link-loaded together), it is proposed that this eventually be done by issuing "remote" procedure calls to "external" procedures in the tools to actually execute commands. This will be accomplished through the Procedure Call Protocal (see Jim White's papers on the PCP). 3a2e3

3) To provide a terminal-independent interface to the tools. 3a2f

Because the CML interpreter handles all terminal interaction, it will present to the tool a small number of virtual terminal classes. Thus, once a tool is developed, little attention need be given to the type or particular characteristics of the terminal the end user may choose to employ while using the tool. In fact, the cost of creating new tools should be considerably reduced because of these facilities. 3a2f1

This means that even though the creators of a tool envisioned the user sitting at a typewriter terminal, the user who happens to be using a display terminal with a pointing device may be able to interact with the tool in a two dimensional sense, pointing to arguments on his screen instead of typing them. 3a2f1a

For tools which wish to make more extensive use of a display terminal if the user has one, the CML interpreter presents primitives for allocating windows on the display and allows the tool to write/delete/move/make invisible items displayed within the windows. 3a2f1b

4) Possible asynchronous operation. 3a2g

In some instances, it may be possible for the execution of the user's commands to be accomplished in parallel with subsequent command specification and execution. This frees the user to do other things while a lengthy command is being executed by a tool. 3a2g1

6) To provide standard mechanisms for presenting status or error conditions to the user. 3a2h

an error should consist of the following: 3a2h1

a human readable error message 3a2h1a

a code indicating whether this error caused the command to be aborted, completed or undefined and whether the tool is now in a state to receive more commands or should be restarted. 3a2h1b

5) Provide the user with enhanced, consistent help facilities while usng any such tool. 3a2i

6) Allow for a common statistics gathering point for analysing user interaction characteristics such as error rates, frequency of issuing given commands or groupings of commands, and average user-observed execution times for commands. 3a2j

7) Provide a convenient way of subsettting the commands available to the user. 3a2k

In the current case, a hypothetical computer was postulated that had as primitives instructions that interacted with a

human user. A "program" for this machine forms a tree-like structure, which we call a Grammar. The two addresses are called the "successor" and "alternative" addresses. The successor address points to another instruction to be executed if this instruction succeeds (i.e. is "TRUE"). The alternative address points to another instruction(s) to process in parallel with this one. That is, a number of instructions are processed in "parallel" such that when any one of the instructions in the current set of alternative instructions succeeds, then the Program Counter is advanced to its successor. That instruction and its alternatives are then processed. 3a3

At any point in time, the machine is attempting (presumably by interacting with a human user) to choose a path through the tree. An illustration might be helpful here. At a certain point in time the Program Counter might point to an instruction to recognize a command word, a "reserved" word in the command language. This instruction might have alternatives which are attempting to recognize other command words. These command words might represent the verbs of commands the user can give to the system or might represent refinements to a command already partially specified. The machine picks a path through these alternative command words, although how this is accomplished is left purposely unspecified. For it is precisely the "HOW" of this machine's path finding that imbodies the human factors considerations and human-machine interaction disciplines, which can, and in our case do, vary from user to user. Thus, how the system interacts with the user is independent of the commands the user has available -- one can be changed while the other remains constant. 3a4

Given the existence of the model for this hypothetical machine, we then developed a formal language (CML) and compiler for this machine. It is through this formal language that the staff of ARC now specifies the user interface for the NLS Knowledge Workshop tool system [akw paper] we have been developing for several years. The form of this language is the chief topic of this paper. The object code produced by the CML compiler is called a control language grammar (or simply a Grammar). 3a5

We have developed and used for several months (on a production basis) a simulation of this computer which we call the CML interpreter. Embodied in this implementation are the principles for human-computer interaction that have evolved through many years of useage and evaluation of NLS and other systems. 3a6

It is this interpreter that interacts with the user to help him

specify commands for the system to execute. It prompts him for the type of input required (if the user want it to), shows him the syntactic form of specific commands on request, shows him his actual alternatives at any point in the specification of a command on request and can invoke a semantic help facility if the user requests. This sematic help is derived from a structured data base provided along with the user interface (CML description and grammar) which attempts to describe in English the intended use of the various commands and the tool as a whole. This data base is highly structured to allow the user to get the information he needs quickly without wading through pages of output[ref rww chapter and help chapter]. 3a7

Thus, a tool now consists of three parts: 1) a CML description of the user interface, 2) a semantic help data base, and 3) an execution module that carries out the commands specifiable by the user. 3a8

To allow the user interface to be individually tailored, we have added a data base called the "user profile" which describes to the CML interpreter how much prompting and feedback the user wants, what recognition scheme he wishes to use to select command word alternatives, and many other ideosyncratic features of the user interface[ref rww chapter]. There is a special set of commands for modifying this data base and consequently the behavior of the system. 3a9

And finally, to facilitate user analysis, we have added a user statistics data base in which the interpreter records which commands were used, whether or not errors were made in the specification of the commands, the execution time of the command, and other statistics. 3a10

We are now involved in a second generation CML system which will provide a Frontend system for the National Software Works (NSW) program sponsored jointly by ARPA and the Air Force. In this second generation system the execution functions that implement the semantics of the commands are called through the Procedure-Call Protocol (PCP) and the Multi-Process Support System [ref jew's pcp doc]. Thus, unlike the current CML system, the execution functions may be written in any language which can be interfaced to PCP. In addition, certain aspects of the language will be improved as discussed in the conclusions below. 3a11

The intention with the NSW as with NLS is to provide the user access to a number of general or specialized tools in such a way that the command discipline he uses remains constant even

though the particular vocabulary changes from tool to tool as appropriate to describe that tool's functions. In the case of the NSW and for future releases of NLS, this user interface will reside not only on a PDP-10 but will also be available on a dedicated Frontend PDP-11 (or other mini computer) for better and less expensive responsiveness. We anticipate that heavily used tools or commands will, in time, actually be executed in the PDP-11. In addition to increased system responsiveness, this will reduce network communication and will afford user's a certain amount of insulation from network or large-computer unavailability. 3a12

As described above, the Frontend system consists of the following: 3a13

1) A formal language (CML) for specifying NSW user interfaces 3a13a

2) A compiler for that formal language that runs under TENEX as a subsystem or from NLS 3a13b

3) Tool Grammars, products of the CML compiler or any other such program 3a13c

4) A CML interpreter that processes a CML grammar in order to work with the user in specifying syntactically correct commands to the NSW. 3a13d

5) A user profile data base that is used by the CML interpreter while interacting with the user. This data base allows the Frontend to be tailored to the individual preferences of the users. 3a13e

6) A user statistics data base, where, if desired, statistics can be accumulated on commands used by a user, error rates, etc. 3a13f

7) Access to a semantic help tool which is employed by the Frontend when the user requests semantic level help with a tool or a command. It is presumed that each tool, in addition to supplying the Frontend with a grammar will also supply it with the name of a help data base whose structure and content, as with the grammar, are the sole responsibility of the tool builder/supplier. 3a13g

This help tool could also be kept informed of the user's dialog with the Frontend and could have access to the

tool grammar, the current parse state of the user, and the user's profile. 3a13g1

The rest of this paper describes in more detail the Control Meta Language and the CML interpreter. 3a14

For a discussion of the NLS command Language, implemented with this system, see chapter [watson's]. 3a15

INTRODUCTION TO THE CONTROL META LANGUAGE 3b

As discussed above, the Control Meta-Language (CML) is a vehicle for describing the syntax of the user interface to application programs. The syntax is described through the tree-meta alternation (denoted by /) and succession (denoted by juxtaposition) concepts[ref tree meta reports]. The semantics are introduced via built-in functions, semantic conventions, and parse functions. 3b1

No attempt is made to describe the full semantics of any command via CML, but it is hoped that the Frontend interface (parsing and feedback operations) may be explicitly accommodated with these facilities. It will still be necessary, and desirable, to use execution functions to perform the low-level semantics of the command. We call the collection of these execution functions and there support routines and data structures the tool "Backend". The CML describes how the command "looks" to the user, rather than what it does inside the tool. 3b2

The CML supportes zero look ahead, phrase structured, context free control languages. 3b3

USE OF CML 3c

The user interface for a tool is defined in the CML specification language. This CML "program" is then compiled by the CML compiler (written using ARC's tree-meta compiler compiler system[ref tree meta reports]) to produce object code (called a Grammar) which is interpreted by a Control Language Interpreter (CLI). The Control Language Interpreter is cognizant of the device dependent feedback and addressing characteristics of the user's terminal through an appropriate interface to a terminal control module described in [CHI's NCC paper]. 3c1

SYNTAX NOTES 3d

The following meta symbols are used in this discussion of the CML: 3d1

.ID	An Identifier	3d1a
.SR	A quoted string	3d1b
$ element	Zero or more occurrences of the following	3d1c
/	Denotes alternatives. A/B means A or B.	3d1d
# element.	At least one occurrence of the following	3d1e
%	brackets comments.	3d1f
()	used for grouping to control precedence.	3d1g
[]	used to denote optional elements	3d1h
'	preceeds literal characters	3d1i
"	encloses literal strings.	3d1j
#<...> element,	At least one occurrance of the following separated by whatever ... represents	3d1k
$<...> element,	Zero or more occurrance of the following separated by whatever ... represents	3d1l

ELEMENTS OF CML 3e

PROGRAM STRUCTURE 3e1

The basic compilation structure of a CML program is described by: 3e1a

```
file        = "FILE" .ID $dcls srule #subsys "FINISH";      3e1a1
```

Explanation: 3e1a1a

The "file" construct brackets the definition of control language subsystems. 3e1a1a1

Declarations of variables, execution and parse

functions, and external identifiers may be made at this level. In addition, global parsing rules may appear here and be invoked in commands by simply specifying their names. 3e1a1a2

```
subsys      = "SUBSYSTEM" ,ID % subsystem handle %
              KEYWORD ,SR  % recognition string %
              #(command / rule) "END.";                    3e1a2
```

Explanation: 3e1a2a

The "subsystem" contruct brackets a set of rules or commands (generally a set of related commands that the command language designer wants to cluster together). Commands beginning with the keyword COMMAND are linked together to form a command language subsystem. 3e1a2a1

```
command     = ("COMMAND" / "INITIALIZATION" /
              "TERMINATION" / "REENTRY") rule ;            3e1a3

rule        =  ,ID '=  exp '; ;                          3e1a4
```

Explanation: 3e1a4a

The subsystem may include a rule preceded by the reserved INITIALIZATION or TERMINATION. If specified, these rules will be executed once upon subsystem initialization/termination, respectively. This enables, for example, a tool to open and initialize a work file when it is started and to close it after the user's last command has been issued. 3e1a4a1

The subsystem may include a rule preceded by the keyword RENTRY which will be executed upon rentry in the subsystem after executing commands in other subsystems. 3e1a4a2

The command Language Interpreter allows the user to freely move among subsystems. Thus, the user may give commands to one subsystem for a while, then give commands to another, and finally return to the first. The RENTRY rule will be executed when the user resumes giving commands to the first subsystem. This might be necessary, for example, to ensure that a work

file or data structures were still in a consistent form. 3e1a4a2a

Each rule/command is named with an identifier. This name may be used as a term in any other rule, indicating that the named rule is to be invoked at that point in the parse. 3e1a4a3

DECLARATIONS 3e2

Declarations are used to associate attributes with identifier names which are used in CML programs. If not declared, identifiers are defined by their first occurrence according to the following rules. 3e2a

1) Identifiers appearing on the left hand side of an assignment statement are defined as "VARIABLES". 3e2a1

2) Identifiers followed by a subscripted list are assumed to be of type "FUNCTION". 3e2a2

3) All other undefined identifiers are assumed to be names of parse rules or commands. 3e2a3

The syntax of the declare statement is given by: 3e2b

```
dcls        = ("DCL" / "DECLARE")
              ( ["VARIABLE" / "FUNCTION" / "PARSEFUNCTION" /
              "EXTERNAL"] #<',> ,ID
              / "EXT-KEYWORD" #<',> ,SR);
```
3e2b1

If a declare attribute is not given, type VARIABLE is assumed. Identifiers which are implicitly defined as type FUNCTION or PARSEFUNCTION are EXTERNAL symbols and will be linked by the loader to externally defined symbols with that name. 3e2c

Semantics of the declare attributes: 3e2d

VARIABLE: 3e2d1

a cell which holds pointers to CML records 3e2d1a

FUNCTION: 3e2d2

arbitrary processing function usually invoked to carry out all or part of the execution of a command 3e2d2a

PARSEFUNCTION: 3e2d3

a function which is used to extend CML. Such a fuunction processes input, and is called in "parsehelp" and "parseqmark" mode to supply a prompt string and a ? string, respectively. 3e2d3a

EXT-KEYWORD 3e2d4

precedes a list of keyword strings (#<',>,SR) and indicates that the named keywords are globally defined elsewhere in the system. 3e2d4a

EXTERNAL 3e2d5

associates an external symbol with the named rule/variable permitting separately compiled programs to reference the named rule/variable. 3e2d5a

RECOGNIZERS 3e3

Keyword Recognition 3e3a

The process of keyword recognition is independent of the description of the keywords for CML. In the CML description, each keyword is represented by the full text of the keyword. The algorithm used to match a user's typed input against any list of alternative keywords is known as keyword recognition, and is a function of the Command Language Interpreter and is independent of the CML description of the command. 3e3a1

Keywords are written in the meta language as upper-case identifiers enclosed in double quote marks optionally followed by a set of keyword qualifiers. 3e3a2

```
keyword = ,SR [ '! #qualifier '! ]
```
3e3a2a

The qualifiers serve to control the recognition process for the keywords and to supercede the system supplied internal identification for the keywords. 3e3a3

```
qualifier       = "NOTT"        % Not available from a
                                  typeWriter terminal %   3e3a3a

                 /"NOTD"        % Not available from a
                                  display terminal %      3e3a3b
```

```
/"L1"          % first level keyword
                (to be recognized by
                its first letter) %     3e3a3c

/.NUM          % explicit value for
                keyword %               3e3a3d
```

If the user has specified that he wants some (supposedly frequently used) command words recognized based on their first letter and the rest only after typing an escape character, the CML interpreter attempts to accommodate him. The command language designer has control over which command words will be available to such a user via first letter recognition through the L1 qualifier. 3e3a4

Selection Recognition 3e3b

Three types of selections are built into CML. They are Destination Selection (DSEL), Source Selection (SSEL), and Literal-typein Selection (LSEL). 3e3b1

The literal-typein selection is used to collect literal typein from the user, although it might also allow him to point to text on his display instead of typing it. 3e3b1a

A destination selection is used to allow the user to select one of several items the tool has presented to him. This can be done by pointing to it using a pointing device at a display terminal or by typing characters which the tool will interpret. For example, a tool may manipulate textual or graphical representations of data stored in a file. The tool might have a delete command and would use a destination selection to allow the user to specify the line in the drawing or the word in the text to delete. Thus, when the tool put the display image on the screen, it did so using primitives in the Frontend that supplied identifiers for elements of the display[ref chi ncc paper]. When the user points to an object on the screen, the identifier for it is returned to the tool. 3e3b1b

a source selection is similar to a destination selection but also allows the user to supply the argument as a literal-typein. 3e3b1c

Basically, these are recognizers which require some

entity type as an argument and they return a data structure which represents the selection. The entity type is obtained either by some previous invocation of the recognition function for some list of keyword entities, or use of the VALUEOF (or #) built in function (see example in Appendix 2). 3e3b2

The DSEL, SSEL, and LSEL functions perform all evaluation and feedback operations associated with the selection operations. The command language designer may define new types of selections and define the data structure that is built as a result of the selection. 3e3b3

```
selection       = ("SSEL"/ "DSEL"/ "LSEL") '( param
');                                                    3e3b3a
```

Command Confirmation 3e3c

The process of command confirmation is represented in CML by a built-in parameterless function. 3e3c1

```
confirm         = "CONFIRM";    % command confirmation
%                                                      3e3c1a
```

simple question answering 3e3d

The process of simple question answering is represented in CML by a built-in parameterless function. 3e3d1

```
answer          = "ANSWER";     % YES/NO answer to a
                                  question (TRUE if YES)
%                                                      3e3d1a
```

Other Recognizers 3e3e

Other recognizers may be added through the use of parse functions as described below. 3e3e1

FUNCTION EXECUTION 3e4

Functions may be invoked at any point in the parse by writing a name of some routine and enclosing a parameter list in parentheses. All functions invoked by the interpreter must obey the groundrules set up for interpreter routines. The actual arguments are passed by address, rather than value, and two additional actual arguments are appended to the head of the argument list. 3e4a

```
control   = ,ID % routine name % '( $<',> param ');          3e4a1
param     = factor          % expression element %           3e4a2
          / "VALUEOF" '( ,SR )    % keyword value %         3e4a3
          / '# ,SR          % same as VALUEOF %             3e4a4
          / "TRUE"          % boolean TRUE value (one) %    3e4a5
          / "FALSE"         % boolean FALSE value (zero) %  3e4a6
          / "NULL";         % null pointer value (zero) %   3e4a7
```

PARSING FUNCTIONS 3e4b

Functions which are declared with the PARSEFUNCTION attribute are assumed to be parsing functions. They are called in "parsehelp" mode (described below) and when so called, are passed the address of a string as a third argument. The parsefunction routine then supplies a prompt string which tells what the parsing function does. (see appendix 3 for example). In addition, the parse function should, in a like manner, be prepared to generate a more verbose help string to be used when the user asks to see his current alternatives and a terse syntax string for when the user asks for the syntax of a command. 3e4b1

FEEDBACK CONTROL 3e5

The feedback control elements of CML are used to provide feedback in addition to the normal feedback generated by the recognizers. This is used to implement additional "noise words" and help feedback. 3e5a

1) adding feedback to the command feedback. 3e5a1

A string may be added to the current command feedback by enclosing the quoted string in angle brackets. 3e5a1a

```
extra feedback = '< ,SR '>;                                 3e5a1a1
```

2) replacing the last string in the command feedback. 3e5a2

If the user's terminal allows, it is possible to replace the last string in the command feedback line by using the string replace facility. This is similar

to (i) above except the previous string in the command feedback is deleted before adding the new string. 3e5a2a

```
replace extra feedback = '<"..." ,SR '>;          3e5a2a1
```

A function is also provided to initialize the command feedback mechanisms and clear the command feedback area. 3e5b

```
clear feedback     = "CLEAR";                     3e5b1
```

EXPRESSION DEFINITION 3e6

CML is an expression languge. Commands are defined to be a single expression and expressions are composed of successive/alternative expression factors. Alternative paths are indicated by the character '/ in the expression. 3e6a

The nesting of expressions may be explicitly defined with parenthesis and brackets are used to delimit optional expression elements. 3e6b

```
exp                = #<'/>alternative;            3e6b1

alternative        = #factor;                     3e6b2

factor             = terminal / '[ exp '] / '( exp ');   3e6b3

terminal           = subname        %id/ assign/ function%
                   / confirm        %command confirmation%
                   / feedback       %noise word feedback%
                   / recognition    %built-in recognizers%
                   / loop;          %looping facility%   3e6b4
```

The looping facility permits repetition of a parse rule until an exit condition is met. 3e6c

```
loop               = "PERFORM" ,ID "UNTIL" '( exp ');   3e6c1
```

The ,ID following the keyword PERFORM is a name of a parsing rule which is to be repeated. This rule is evaluated and then the expression following the UNTIL keyword is evaluated. If the expression returns TRUE, then the loop is exited and the next factor in the rule is evaluated. If the expression returns FALSE, then the parse is backed up to the head of the PERFORM, and the named rule is invoked once again. 3e6c2

COMPLETE FORMAL SYNTAX OF CML 3f

```
file            = "FILE" .ID $dcls $rule                                3f1

                  #subsys "FINISH";                                     3f2

subsys          = "SUBSYSTEM" .ID % subsystem handle %                  3f3

                  "KEYWORD" .SR % recognition name %                    3f4

                   #(command / rule) "END.";                            3f5

command         = ("COMMAND" / "INITIALIZATION" /
"TERMINATION"                                  / "RENTRY") rule ;       3f6

rule            = .ID '=  exp '; ;                                      3f7

dcls            = ("DCL" / "DECLARE")                                   3f8

                  ( ["VARIABLE" / "FUNCTION" / "PARSEFUNCTION" /
                     "EXTERNAL"] #<',> .ID
                  / "EXT-KEYWORD" #<',> .SR);                           3f9

exp             = #<'/>alternative;                                     3f10

alternative     = #factor;                                              3f11

factor          = terminal/ '( exp ')/ '[ exp '];                       3f12

terminal        = subname/ confirm/ answer/ feedback/
                  recognition/ loop;                                    3f13

subname         = .ID  [ '_ param/ '( s<',>param ')];                   3f14

confirm         = "CONFIRM";                                            3f15

answer          = "ANSWER";                                             3f16

recognition     = keyword/ builtinrec;                                  3f17

keyword         = .SR [ '! #qualifier '! ];                             3f18

qualifier       = "NOTT"/ "NOTD"/ "L1"/ .NUM;                           3f19

builtinrec      = (("SSEL"/ "DSEL"/ "LSEL") '( param '));               3f20

feedback        = "CLEAR"/ '< ["..."] .SR '>;                           3f21

control         = .ID '( s<',>param ');                                 3f22
```

```
param          = factor/ ("VALUEOF" '( ,SR ') / '# ,SR)          3f23
               /"TRUE"/ "FALSE"/ "NULL";                         3f24
loop           = "pERFORM" ,ID "UNTIL" '( exp ');                3f25
```

THE OBJECT CODE PRODUCED BY THE CML COMPILER -- THE GRAMMAR 4

Each instruction of the object code consistes of the following fields: OPCODE, SUCCESSOR, ALTERNATIVE, ADDR, CTL, and VAL. 4a

The ALTERNATIVE and SUCCESSOR fields 4b

These contain the addresses of an alternative instruction to execute in parallel with the current one and the address of the instruction to execute should this one succeed. Null paths are indicated by 0 valued pointers. 4b1

The OPCODE, ADDR, CTL, and VAL fields 4c

OPCODE is an operation code. CTL contains control bits used by the interpreter (reflecting the NOTD, NOTT, and L1 qualifiers). VAL contains an integer token or zero. ADDR is the address or principal value for the function. 4c1

Possible OPCODES 4d

RECOGNIZERS 4d1

KEYOP -- keyword recognition. 4d1a

CTL = control bits for level 1 commands, Display commands, and TNLS commands. 4d1a1

ADDR = address of keyword literal string 4d1a2

The current input text is matched against the keyword string specified by the current node and all alternatives of the current node. This function performs keyword recognition on all of the alternative nodes of the current node simultaneously. 4d1a3

This function cannot fail. Control remains in the keyword recognition function until appropriate input is recognized or until the control is abnormally wrested via backup or command delete functions. 4d1a4

The value returned in the argument record is a single

word containing the address of the string corresponding to the keyword actually recognized. 4d1a5

CONFIRM -- process command confirmation characters 4d1b

This function interrogates the input text for one of the command confirmation characters. Control remains in this routine until a proper confirmation is recognized, and command termination state is appropriately set. This function always returns TRUE. 4d1b1

The value returned is a single word containing a command completion code which identifies the completion mode. 4d1b2

ANSWER -- process yes/no question answer 4d1c

This function interrogates the input text for one of the yes/no question answer characters. Control remains in this routine until a proper response is recognized. 4d1c1

SSEL -- get a source selection 4d1d

ADDR = not used 4d1d1

The sselect routine is invoked to process a source type selection. The return record generally contains two text pointers which delimit the selected entity (see the appendix for detailed layout of the records returned by the selection recognizers). 4d1d2

DSEL -- get a destination selection 4d1e

ADDR = not used 4d1e1

The dselect routine is invoked to process a destination type selection. The return record generally contains two text pointers which delimit the selected entity (see the appendix for detailed layout of the records returned by the selection recognizers). 4d1e2

LSEL -- get a literal selection 4d1f

ADDR = not used 4d1f1

The lselect routine is invoked to process a literal type selection. The selection type is passed as an actual argument. The return record generally contains two text pointers which delimit the selected entity (see the

appendix for detailed layout of the records returned by the selection recognizers). 4d1f2

VIEWSPECS -- process viewspecs information 4d1g

The viewspec input routine is called to process the input stream for viewspec characters. The return record contains the two updated viewspec control words. This function always returns TRUE. 4d1g1

LEVADJ -- process level adjust information 4d1h

The level adjust input routine is called to process the input stream for level adjust characters. The return record contains a single word which indicates the relative level adjust value (u = +1, d = -1, etc). This function always returns TRUE. 4d1h1

CONTROL FUNCTIONS 4d2

EXECUTE -- transfer of control to another point in the tree. 4d2a

ADDR = address of root of tree for transfer of control 4d2a1

The current point in the tree is marked and control is transferred to the node pointed to by the address field. Control remains in the descendent node until it has been completely parsed, at which time control returns to the successor of the EXECUTE node. 4d2a2

CALL -- subroutine invocation 4d2b

VAL = number of actual parameters 4d2b1

4d2b2

ADDR = address of the subroutine 4d2b3

The appropriate number of actual arguments are popped off of the evaluation stack and passed to the routine whose address is contained in ADDR. 4d2b4

The result from this routine is pushed onto the eval stack if it returns TRUE. 4d2b5

PFCALL -- parsing function invocation 4d2c

VAL = number of actual parameters 4d2c1

ADDR = address of the subroutine 4d2c2

The appropriate number of actual arguments are popped off of the evaluation stack and passed to the routine whose address is contained in ADDR. 4d2c3

The result from this routine is pushed onto the eval stack if it returns TRUE. 4d2c4

This function is also called in "parsehelp" mode to find out what it does. 4d2c5

OPTION -- test for an optional construct. 4d2d

If the next input character is the OPTION select character, then it is read and control is transferred to the node at address ADDR. If the next character is not the OPTION character, then control passes to the successor path of the current node. 4d2d1

FEEDBACK ELEMENTS 4d3

FBCLEAR -- clear the contents of the feedback buffers. 4d3a

The feedback state information and command feedback line are set to their initial or empty position. 4d3a1

ECHO -- appends a noise-word string to the command feedback link 4d3b

ADDR = address of the text string to be appened 4d3b1

RECHO -- replaces the last noise-word string in the command feedback line 4d3c

ADDR = address of the text string which is to replace the last item in the command feedback buffer 4d3c1

VALUE MANIPULATIONS 4d4

LOAD -- loads a pointer to an argument record into the top of the eval stck. 4d4a

ADDR = address of the variable containing the pointer to the argument record. 4d4a1

The pointer value contained in the variable whose address

is contained in ADDR is pushed onto the top of the eval stack. 4d4a2

STORE -- saves a pointer to an argument record in a variable 4d4b

ADDR -- address of the variable 4d4b1

The address of an argument record is fetched from the top of the eval stack and is saved in the variable at address ADDR. 4d4b2

ENTER -- enters a constant value into the argument record pointed to by the top of the eval stack. 4d4c

ADDR -- value to be entered (18 BITS only) 4d4c1

The value is taken from the ADDR field of the instruction and is entered into the argument record for the ENTER node in the path stack (whose address is at the top of the eval stack). 4d4c2

FLOW OF CONTROL IN THE INTERPRETER 5

At any point in the process of parsing, the control pointer for the interpreter points to a structure word in the grammar. A path stack also exists which shows the nodes from which TRUE returns have been achieved. Some operations mark the path stack for halting the backup process. The parser has 4 distinct control states defined as follows: 5a

1) parsing: recognition state where input text is compared with gramatical constructs to determine the parsing path in the parse tree. 5a1

2) backup: A FALSE return has been obtained from some execution/recognition function. The path stack is backed up until a non-NULL alternative path is found, at which time the parse mode is set to parsing, and recognition of the alternative path is attempted. If no non-NULL alternative path is found, then the parse fails and the interpreter returns FALSE. 5a2

3) cleanup: A terminal parse has been achieved and control is passed to each execution routine to reset any state informations set by the routine. 5a3

4) parsehelp: (used only with parsefunctions) Before

calling a parsefunction in "parsing" mode, the function is called in "parsehelp" mode to solicit a user prompt string. 5a4

The general flow of control is: 5b

1) An initial path stack entry is constructed, and the parse mode is set to parsing. The execution function for the current node is evaluated. A pointer to the "function state record" is passed to the routine. The state record contains the return values for the function as well as a record of any state information saved by the function (for backup purposes). 5b1

2) A prompt string is generated for the user indicating in a terse fashion what his current alternatives are. If he wishes expansion on this he may ask for his current alternatives or for the syntax of the rest of the command. 5b2

3) If the function returns TRUE, then the successor to the current node becomes the current node. If this is NULL, then the ptrstk stack is backed up until a non-NULL successor path is found. If none is located before the bottom of the current parse state is reached, then the root of a parse tree has been reached, and a command has been successfully executed. In this case the command reset operation is performed and the interpreter is set to "parsing" mode once more. 5b3

4) If the function returns FALSE then the parser mode is set to "backup" and a non-NULL alternative path is sought. 5b4

After a command has been executed, the parsing path for the tree is re-evaluated in "reverse order" beginning with the terminal node of the path. Each execution function is re-invoked, in "cleanup" mode, and is passed the handle for the state information record which it generated on the forward pass through the grammar. Each execution routine has the responsibility of resetting any state information which it wishes to do at the termination of a command. Cleanup continues until a "starting point" is reached in the parse. This is generally the beginning of the command. At this point, the interpreter "shifts gears" and goes into forward or recognition mode and begins back down the grammar for the language. 5c

The same backup mechanism is also used during command specification in order to back up the parse to allow the respecification of all or part of the command. The command delete function backs out of the parse tree until the beginning of the command is reached. 5d

The same backup mechanism may be adapted to control the partial backup required for executing commands in "repeat mode" where at least one of the alternatives are defaulted to their current values. 5e

PROBLEMS WE HAVE ENCOUNTERED WITH CML 6

The principle problem we have encountered is that some of the recognizers (keyword recognition, command confirmation, LSEL, SSEL, DSEL, and so forth) cannot fail. This is purely an implimentation decision that was made regarding the CML Interpreter and, consequently, does not impact the language itself. In addition, the CML Interpreter was implimented as a stack machine and would better serve our needs as a machine with an accumulator and an argument stack. 6a

Also, the manner in which the user-input prompt, the current alternatives, and the syntax of commands is generated should be more standardized to avoid some of the problems and anomallies we have encountered to date. These problems have chiefly been caused by the knowledge the interpreter has of some functions and lack of knowledge about others. 6b

In order to serve the needs of a wider range of tools (application programs) we feel that the declaration facility of CML should be expanded to allow the command language designer to define how to handle many special things such as collection of parameters of a form specific to the tool. In addition, we would like to make the CML Interpreter system available through an interface [ref white's pcp papers] that does not require the tool to be written in the same language as the Interpreter. 6c

APPENDIX 1: USING THE CML SYSTEM [Should we delete this section?] 7

WRITING CML PROGRAMS 7a

Source programs for the CML compiler are free form NLS or TENEX sequential files. Comments may be used wherever a blank is permitted and the structural nesting of the source file is ignored by the compiler. 7a1

COMPILING CML PROGRAMS 7b

CML source programs are compiled into REL files with the Compile File command in the PROGRAMS subsystem. CML is the compiler name for the CML compiler. 7b1

RUNNING CML PROGRAMS 7c

A complete interactive subsystem usually consists of three distinct parts: (1) The syntactic description for the subsystem command language, (2) The parser interface routines ("X" level parsing support routines), (3) Core execution functions. 7c1

If a CML subsystem is to be run as a user program, then the rel files for the syntax, parsing support, and execution functions are loaded into the user programs buffer with the Load Program command. 7c2

After loading the rel-files the user's subsystem is connected to the set of available subsystems with the Attach Subsystem command. The name specified in this command is the name of handle for the subsystem (the .ID appearing on the SUBSYSTEM statement of the CML program). 7c3

The user's subsystem may then be invoked by using the GOTO command, as the system will now know about the new subsystem. 7c4

FUNCTION INTERFACE PROTOCAL 7d

The syntax of the function call in the CML meta-language is similar to that of most programming languuages: the name of the function is followed by a list of expressions enclosed in parenthesis. In the CML system however, there are some strict rules which apply to all execution functions invoked by the interpreter. These rules are enumerated below: 7d1

1) Additional actual arguments 7d1a

Preceeding any actual arguments which appear in a function reference in CML, the interpreter supplies two additional actual arguments. These are: 7d1a1

1) a pointer to the "function state record" 7d1a1a

2) an integer which defines a parsing mode 7d1a1b

= parsing: normal execution mode 7d1a1b1

= backup: backup after a FALSE path is taken 7d1a1b2

= cleanup: resetting of state after completion 7d1a1b3

of command 7d1a1b4

= parsehelp: soliciting prompts string (parse 7d1a1b5

functions only) 7d1a1b6

These addtional arguments must be used by all execution functions to determine what they are to do. The pointer to the "function state record" is used to return values from the function and to save state information associated with a particular invocation of the function. The length of the function state record is presently 10 words and this record may be formatted in any manner appropriate to the function. 7d1a2

If 10 words is not sufficient space to record all of the state associated with a particular invocation of a function, then the function must use a storage allocator to allocate the additional storage and record the handles to the allocated storage in the function state record. Note that if this additional "local state" storage is required, then it is the responsibility of the execution function to de-allocaate the local state storage when called in backup or cleanup modes. 7d1a3

2) Returning parse failure 7d1b

All execution functions are passed a pointer to their function state record. If the function processes normally, then it returns the same pointer as its only return value. If the function decides that the parse should fail at a given point, then it returns FALSE. 7d1b1

3) Passing arguments by address 7d1c

All of the actual arguments in a function call on an execution function are passed by address rather than by value. The values actually passed are pointers to the function state records corresponding to the actual arguments. The format of the function state records are defined by the execution functions which manipulated them, and thus the location of parameter values in these records is determined by convention, the caller and callee having previously agreed to a particular layout for the function state record. The layout of the records for the built-in interpreter functions in given elsewhere in this appendix. 7d1c1

4) Order of control 7d1d

An execution function will always be called in parsing mode before it is called in backup or cleanup modes. 7d1d1

A function routine which saves state information in the function state record must initialize its state record to some consistent state before it calls any subroutines which may cause SIGNALS or otherwise cause control to abnormally pass above the execution funtion. 7d1d2

APPENDIX 2: SAMPLE CML PROGRAM 8

```
FILE nlsexample                                                    8a

    SUBSYSTEM nlseditor KEYWORD "BASE"                             8a1

      % COMMON RULES %                                             8a1a

        % PARAMETER TYPE DEFINITIONS %                             8a1a1

          editentity = textent / structure;                        8a1a1a

          % TEXT PARAMETER TYPE DEFINITIONS %                      8a1a1b

            textent =                                              8a1a1b1

              "CHARACTER" / "WORD" / "VISIBLE" / "INVISIBLE" /
              "TEXT" / "LINK" / "NUMBER";                          8a1a1b1a

          % STRUCTURE PARAMETER TYPE DEFINITIONS %                 8a1a1c

            structure = "STATEMENT" / "GROUP" / "BRANCH" /
            "PLEX";                                                8a1a1c1

      COMMAND %replace%                                            8a1b

        zreplace =                                                 8a1b1

          "REPLACE"                                                8a1b1a

            type _ editentity                                      8a1b1a1

              % The rule EDITENTITY defined above is
              evaluated. The one chosen (via user input) is
              stored in the variable TYPE. %                       8a1b1a1a

            <"at"> destination _ DSEL(type)                        8a1b1a2

              % The usee is presented the noise word "at" and
              requested to supply a destination of the type
              chosen from EDIIEETITY. The user must then
              identify the item to be replaced. The
```

```
            representatioo of this item is stored in the
            variable DESTINATION. %                          8a1b1a2a

         <"by"> source _ LSEL(type)                          8a1b1a3

            % The replacement is collected from the user and
            stored in the variable SOURCE. %                 8a1b1a3a

         CONFIRM                                             8a1b1a4

            % Have the user confirm that he wants the
            replacement to take place as specified. %        8a1b1a4a

         xreplace( type, destination, source );              8a1b1a5

            % call the primitive in the aaplication program
            that performs replacements. Pass it the type of
            thing to replace, the specific instance of that
            type to be replaced, and the replacement. %      8a1b1a5a

      COMMAND %load%                                         8a1c

         zload =                                             8a1c1

            "LOAD"                                           8a1c1a

               type _ ("FILE"/"PROGRAM")                     8a1c1a1

                  % this command allows user's structured text
                  files and programs to be loaded into NLS for
                  user manipulation and execution, respectively. %
                                                           8a1c1a1a
               filename _ LSEL(#"OLDFILENAME") CONFIRM       8a1c1a2

                  % Collect the name of an old file from the user.
                  The file may be the one to load or it may
                  contain the program to be link-loaded. %   8a1c1a2a

               xload(type, filename);                        8a1c1a3

                  % pass the appllcation program's load primitive
                  the type of load and the file name. %      8a1c1a3a

      COMMAND %interrogate user to help him send mail to other
      users%                                                 8a1d

         interrogateCmd =                                    8a1d1
```

```
"INTERROGATE"                                                    8a1d1a

   CONFIRM                                                       8a1d1a1

      % User wants to be interrogated for needed info
      to send mail to other users. %                             8a1d1a1a

   CLEAR <"distribute for action to:">                           8a1d1a2

      content _ LSEL(#"IDENTLIST")                               8a1d1a2a

      setfield(#"ACTION", content)                               8a1d1a2b

      % CLEAR causes Carriage Return Line Feed on a
      typewriter-like terminal and causes the command
      area to be cleared on a display.  The
      application function setfield is called to set
      the "action" field in the current message header
      to the list of user-recipients supplied by the
      user and stored in CONTENT. %                              8a1d1a2c

   CLEAR <"distribute for information-only to:">                 8a1d1a3

      content _ LSEL(#"IDENTLIST")                               8a1d1a3a

      setfield(#"INFORMATION", content)                          8a1d1a3b

   CLEAR <"title:"> content _ LSEL(#"TEXT")                      8a1d1a4

      setfield(#"TITLE", content)                                8a1d1a4a

   CLEAR <"type of source:">                                     8a1d1a5

      (                                                          8a1d1a5a

         "MESSAGE" type _ #"STATEMENT"                           8a1d1a5a1

            content _ LSEL(#"TEXT")                              8a1d1a5a1a

               % Message is the same as statement. %
                                                                 8a1d1a5a1a1
         / type _ "FILE"                                         8a1d1a5a2

            content _ DSEL(#"CHARACTER")                         8a1d1a5a2a

               % The user may specify any character in
               the file. %                                       8a1d1a5a2a1
```

```
                / type _ structure                                  8a1d1a5a3

                   <"at"> content _ SSEL(param)                     8a1d1a5a3a

                      % Since this is an SSEL, the user may
                      type it or specify its location in one
                      of his files. %                               8a1d1a5a3a1

                / type _ "OFFLINE" <"document">                     8a1d1a5a4

                   <"located at"> content _ LSEL(#"TEXT")           8a1d1a5a4a

                      % If it is an offline hardcopy
                      document, simpley have the user descibe
                      where it is being stored. %                   8a1d1a5a4a1

                )                                                   8a1d1a5a5

             setfield(type, content)                                8a1d1a5b

          CLEAR <"show status?"> (ANSWER showstatus() /
          DUMMY)                                                    8a1d1a6

             % If the user answers "YES", call SHOWSTATUS to
             present the current specification of the mail to
             the user. %                                            8a1d1a6a

          CLEAR <"send the mail now?"> (ANSWER xdoit() /
          DUMMY) ;                                                  8a1d1a7

             % If the user answers "YES", call XDOIT to send
             the mail as specified, otherwise simply let him
             use other commands to change the specifications
             and send it. %                                         8a1d1a7a

    END.                                                            8a1e

FINISH                                                              8a2
```

APPENDIX 3: SAMPLE INTERPRETER PARSEFUNCTION ROUTINE 9

Assume that in some command we want the typein of a number to appear as an alternative of some set of keywords. We can accomplish this by defining a parsefunction (call it looknum) which looks at the next input character and succeeds if the next character is a digit and fails otherwise. If we write this function as the first alternative in some command, then control

```
will pass from the interpreter to the  parsefunction before it
passes to the keyword interpreter.                                    9a

Suppose our command looks like:                                       9b

   COMMAND sample =                                                   9b1

      "INSERT"                                                        9b1a

         % determine the type of insert %                             9b1a1

            ( looknum() <"number">  type _ #"NUMBER"                  9b1a1a

            /  type _ ( "TEXT" / "LINK") )                            9b1a1b

         % the variable TYPE now contains  NUMBER, TEXT, or LINK,
         We now use the LSEL function to get a selection of this
         type and store it in the variable SOURCE %                   9b1a2

            source _ LSEL( type)                                      9b1a2a

         % get a command confirmation to make sure user wants this
         done %                                                       9b1a3

            CONFIRM                                                   9b1a3a

         % now invoke the insert execution function passing as
         arguments the entity type and the selection of that type
         %                                                            9b1a4

            xinsert( type, source);                                   9b1a4a

Now take a look at the parsefunction looknum which is called by
the interpreter both when prompting the user and also during the
actual parse of the command .                                         9c

   % LOOK FOR A NUMBER %                                             9c1

      (looknum) PROC(                                                 9c1a

         % looknum looks at the next input character, if it is a
         digit, then a true return is taken else FALSE is returned
         %                                                            9c1a1

         % FORMAL ARGUMENTS %                                        9c1a2

            resultptr,      % ptr to the function state record %     9c1a2a
```

```
        parsemode,      % parsing mode for the interpreter %    9c1a2b
        string);        % ptr to prompting string %             9c1a2c
    REF resultptr, string;                                      9c1a3
    %------------%                                              9c1a4
    CASE parsemode OF                                           9c1a5
      = parsing:                                                9c1a5a
        CASE lookc() OF                                         9c1a5a1
          IN ['0, '9]:                                          9c1a5a1a
            NULL;                                               9c1a5a1a1
        ENDCASE RETURN (FALSE);                                 9c1a5a2
      = parsehelp:                                              9c1a5b
        *string* _ "NUM:";                                      9c1a5b1
    ENDCASE;                                                    9c1a6
    RETURN (&resultptr);                                        9c1a7
    END.                                                        9c1a8
                                                                9c1a9
```

References: 10

1) 68 report 10a

2) display interaction report 10b

3) other old report 10c

1) akw paper 10d

2) akw terminal specs paper 10e

6) ncc paper 10f

CHI 27-FEB-75 12:16 25504

The Control Meta Language -- A Formal Language for Describing and Implementing Interactive Control Languages

(J25504) 27-FEB-75 12:16;;;; Title: Author(s): Charles H. Irby/CHI; Distribution: /DVN([ACTION]) ; Sub-Collections: SRI-ARC; Clerk: CHI; Origin: < IRBY, CML-FINAL,NLS;3, >, 27-FEB-75 12:05 CHI
;;;; updates to (ijournal, 20438,)####;

Preliminary Bit Eight Format Specification

THE PCPB8 FORMAT 1

Introduction 1a

Data structures may be encoded according to PCPB8 when the physical channel allows messages which are streams of 8-bit binary bytes. 1a1

The first byte of a data structure is a type code, with the type zero having the special interpretation indicating that a key is present for this data structure, non-zero codes indicate element types. 1a2

Data Structure Encoding 1b

Key 1b1

FLAG (1 byte) 1b1a

VALUE (any element) 1b1b

Elements 1b2

CHARSTR 1b2a

TYPE (1 byte) 1b2a1

SIZE (1 byte) 1b2a2

COUNT (size bytes) 1b2a3

TEXT (count bytes) 1b2a4

BITSTR 1b2b

TYPE (1 byte) 1b2b1

SIZE (1 byte) 1b2b2

COUNT (size bytes) 1b2b3

BITSTRING (Count bits ((count+7)/8) bytes) 1b2b4

INTEGER (two's complement) 1b2c

TYPE (1 byte) 1b2c1

SIZE (1 byte) 1b2c2

COUNT (size bytes)	1b2c3
INTEGER (count bits ((count+7)/8) bytes)	1b2c4
BOOLEAN	1b2d
TYPE (1 byte)	1b2d1
VALUE (1 byte)	1b2d2
FALSE=0	1b2d2a
TRUE =1	1b2d2b
EMPTY	1b2e
TYpE (1 byte)	1b2e1
LIST	1b2f
TYPE (1 byte)	1b2f1
SIZE (1 byte)	1b2f2
COUNT (size bytes)	1b2f3
REPEAT (1 byte)	1b2f4
SPECIFIEDELEMENTS=0	1b2f4a
Count Data Structures	1b2f4a1
REPEATEDELEMENT=1	1b2f4b
One Data Structure (representing count repeated instances)	1b2f4b1
REPEATEDHEADER=2	1b2f4c
One Data Structure Header	1b2f4c1
Count Data Structure Values	1b2f4c2
INDEX	1b2g
TYPE (1 byte)	1b2g1
VALUE (1 byte)	1b2g2

The value represents an integer in the range 1 through 255 1b2g2a

Data Structure Format 2

```
               *----*-----------*
   datastruc   * Key * element *
               *----*-----------*
                                                                  2a

      key                                                        2a1

                 *-----*-----------*
         key     *  0  * element *
                 *-----*-----------*
                    1       x                                    2a1a

      element                                                    2a2

                 *-----*------*-------*------*
         charstr *  1  * size * count * text * Network ASCII
                 *-----*------*-------*------*
                    1     1     size    count                    2a2a

                 *-----*------*-------*------*
         bitstr  *  2  * size * count * bits *
                 *-----*------*-------*------*
                    1     1     size    count   ((count+7)/8 bytes)   2a2b

                 *-----*-------*---------*
         integer *  3  * count * integer * twos complement
                 *-----*-------*---------*
                    1     1       count                          2a2c

                 *-----*--------*
         boolean *  4  * 0 or 1 * 0 for FALSE or 1 for TRUE
                 *-----*--------*
                    1     1                                      2a2d

                 *-----*
         empty   *  5  *
                 *-----*
                    1                                            2a2e

                 *-----*------*-------*--------------------------*
         list    *  6  * size * count * count-data-structures *
                 *-----*------*-------*--------------------------*
                    1     1     size            count            2a2f
```

```
          *-----*--------*
   index  *  7  * value  * small positive integer 0<value<256
          *-----*--------*
             1       1
```

2a2g

Examples 3

Character string "ABCDE" 3a

```
*-----*-----*-----*-----*-----*-----*-----*
*  1  *  1  *  5  *  A  *  B  *  C  *  D  *  E  *
*-----*-----*-----*-----*-----*-----*-----*
```

3a1

Bit string "100011111101011" 3b

```
*-----*-----*-----*----------*----------*
*  2  *  1  * 1100 * 10001111 * 10101100 *
*-----*-----*-----*----------*----------*
```

3b1

Integer "-3" 3c

```
*-----*-----*----------*
*  3  *  1  * 11111101 *
*-----*-----*----------*
```

3c1

Boolean "TRUE" 3d

```
*-----*-----*
*  4  *  1  *
*-----*-----*
```

3d1

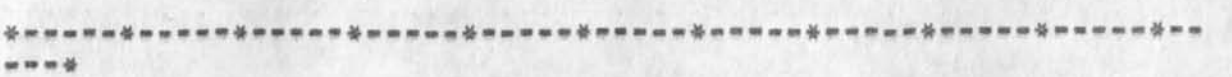
Empty 3e

```
*-----*
*  5  *
*-----*
```

3e1

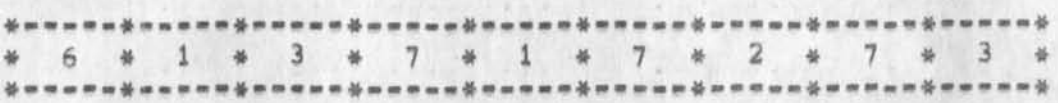
List of a character string "ABC" and a boolean "FALSE" 3f

```
*-----*-----*-----*-----*-----*-----*-----*-----*-----*-----*-----*-----*-----*-----*
*  6  *  1  *  2  *  1  *  1  *  3  *  A  *  B  *  C  *  4  *  0  *
```



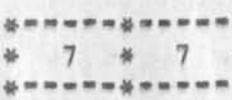
3f1

List of three the indexes "1", "2", "3" 3g



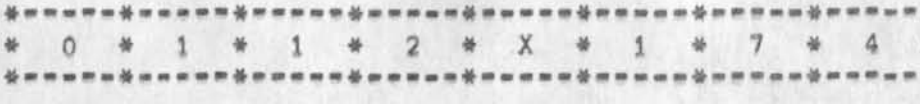
3g1

Index "7" 3h



3h1

Index "4" with character string Key "X1" 3i



3i1

Preliminary Bit Eight Format Specification

(J25505) 27-FEB-75 17:36;;;; Title: Author(s): Jonathan B. Postel/JBP; Distribution: /JEW([INFO-ONLY]) DSM([INFO-ONLY]) ; Sub-Collections: SRI-ARC; Clerk: JBP; Origin: < POSTEL, PCPB8.NLS;7, >, 27-FEB-75 17:34 JBP ;;;;####;

DCE JCN 27-FEB-75 17:49 25506

An ARC IR&D Proposal: AKW Technology Transfer Techniques

Submitted and accepted in Jan 75 for an SRI IR&D Program

An ARC IR&D Proposal: AKW Technology Transfer Techniques

Principal Investigator: James C Norton 1

Basic/Applied: Applied 2

Expenditures: 1975 (proposed) 3

Labor: $75K 3a

Other: $80K (Computer services and terminal lease, for two ssots and work stations devoted to this practice, coaching, materials development, etc.) 3b

Total: $155K 3c

Key words: Technology Transfer, Online Information Systems, Augmentation 4

Potential client relationship: 5

X DoD 5a

X NASA 5b

X Other government agency (almost any) 5c

X Other client category (almost any) 5d

OBJECTIVE 6

The problem is to introduce into external institutions radical innovations representative of the way in which their future knowledge work will be done. Over the past ten years of concentrated, single-purpose work, involving over $10 million of govenment R&D support, ARC has developed an extensive, coherent system of tools and techniques to support an Augmented Knowledge Workshop, incorporating computer and communication tools so advanced that their replication within an application environment would represent extreme barriers in cost, qualified implementation personnel, qualified applications coaching and training personnel, suitably knowledgeable and oriented decision makers, etc. 6a

Without first-hand, real-work experience with such tools, potential users are unable to perceive the possibilities, problems and potential value; they could not provide adequate justification for the very high implementation costs of acquiring computer and communication support facilities within their own organization, and they would either have to accept a plan for new worker organization, methods, procedures, etc. as

designed for them by a outsider or base their plans on untutored surmise of inexperienced internal staff. 6a1

Over the past five years ARC has planned for an experimental, relatively large-scale program aimed toward transferring this type of technology into external institutions. For the past year we have operated a pilot service that external clients can subscribe to, where the computer-communiations tools are brought into the clients' home sites via special communication channels from a central computer facility operated by a commercial time-sharing company under contract to us. We now have about $920K/yr subscription business, and this is expected to increase perhaps three-fold during the coming year. Many government agencies are perceiving direct value from the service, and the scope of exploratory applications is rapidly expanding. 6b

In the first year of service, we have learned quite a bit about the human support component of the service required by a client in order to facilitate the introduction of these tools into their exploratory application operation, and to build awareness, perspective of applicability potential, to overcome inappropriate fears, prejudices and expectations, to evolve skills and knowledge toward increased knowledge-worker effectiveness, to learn how to provide sensible cost/payoff exploration plans for their management, to choose personnel and exploratory applications sensibly, etc. 6c

The importance of these human support services is so high that our further stages of this technology-transfer experiment are planned to include a significant increase in the staff of transfer specialists. These specialists require a kind of training and experience that we as yet don't know how to provide quickly and effectively -- we've depended so far on using staff that have already acquired experience in developing and using these tools and techniques, and that have an aptitude for dealing with people. We have to depend in the future upon hiring and training new people for these transfer-facilitation roles. 6d

The objective of this IR&D Program is thus threefold: 6e

1) Develop ARC technology-transfer staff to the point where they can directly contribute to supporting the transfer process into our client institutions. 6e1

2) Develop the materials and methods that regularize the training, so that it will be possible for instance to transfer this training capability into a client organization. 6e2

3) Ready our capability to run courses aimed at training AKW users or AKW-user trainers. 6e3

APPROACH 7

Develop a training curriculum, with appropriate materials, practice exercises, testing procedures, etc. (requiring time of current, experienced staff, plus work-station access) 7a

After screening and recruiting staff with appropriate backgrounds and aptitudes, this program would support the experienced-staff trainers and the new-staff trainees. 7b

The opportunity seems to exist for offering one-shot, fixed-price training courses as a significant component of this technology-transfer process. If our experience in the early stages of this program indicate its feasibility, we would prepare one or more trial courses. 7c

FUTURE PLANS 8

To work toward the objectives and approach outlined above during 1975. 8a

DCE JCN 27-FEB-75 17:49 25506

An ARC IR&D Proposal: AKW Technology Transfer Techniques

(J25506) 27-FEB-75 17:49;;;; Title: Author(s): Douglas C. Engelbart, James C. Norton/DCE JCN; Distribution: /RWW([INFO-ONLY]) JCN([INFO-ONLY]) JHB([INFO-ONLY]) RA3Y([INFO-ONLY]) RLL([INFO-ONLY]) SGR([INFO-ONLY]) ; Sub-Collections: SRI-ARC; Clerk: DCE;

DCE 27-FEB-75 18:12 25507

ARC Planning Notes from 3 Nov 74

For the record; un-edited; as used in planning dicussions

ARC Planning Notes from 3 Nov 74

NOTE** This draft is for SRI-ARC, and other interested SRI personnel, to look over and comment upon. I would appreciate discussion. I'll return from Washington on Monday 11 Nov, and would like to talk these matters over with whomever is interested. DCE. 1

ARC has just completed its proposed budget for CY 75 and 76. This memo summarizes my understanding of the conditions associated with this two-year budget, and the framework of strategy, policy, and intra-ARC agreemnts. 2

Note: In the text below, "ADAG" and "AAG" stand respectively for ARC Development and Analysis Group, and ARC Applicatons Group (headed respectively by Dick Watson and Jim Norton). 2a

Business-planning framework, within SRI 3

ARC faces extreme challenges in pursuing the potentials of its AKW-Systems Community Development. For what would be a reasonable approach toward these challenges, taken from a distant view of the potentials (social payoff, timeliness, resources apparently ready to unlock, etc.), the commitments would be far beyond the physical capabilities of ARC staff, and far beyond the elasticity of SRI's financial operations. 3a

ARC can only ask of SRI that a reasonable framework be provided, in which ARC can plan and pursue these challenges within guidelines that account suitably for SRI's relatively fragile financial position, for SRI's potential gains in business-growth areas of importance to it, and for ARC's need to do investment/payoff planning over longer than one-year budgeting cycles. 3b

I think that we are approaching a framework that is quite satisfactory to ARC. Let me describe it below in my own terms, as basis for further discussion leading to a firm, mutually understood agreement. 3c

[NOTE: This is the framework within which ARC prepared its proposed 2-year budget -- just submitted. It has been worked out with Bart Cox in some detail, but it remains to be negotiated among divisions and within the financial problems and policies of SRI's top executives. We hope that it works out.] 3c1

First, we established a "Baseline Income Level" for ARC, based upon its current Total Payroll and a negotiated Efficiency Factor -- where we hope that it would be satisfactory to SRI if ARC maintained this income level over the CY 75-76, two-year period, assuming constant ARC Total Payroll. From this approach we

calculated the Net Two-Year Baseline Accumulated Income (TYBAI), i.e, what ARC would contribute to SRI's income over this two year period if we operated in a satisfactory, constant-size basis. 3d

Then, the basic agreement that we seek to establish with SRI is that ARC commit every effort toward producing that TYBAI. SRI's agreement with ARC would be that we can work with a budget plan in which our size may increase significantly, our contribution to SRI income in 75 may be considerably below a one-year accrual at the Baseline level, but where there is second-year increase in total revenue and in efficiency such that we can recapture the deficit and meet the agreed-upon two-year accumulated-income target. 3e

We assume that limits would be put on the excursions "into debt" of various sorts would be set; and/or that periodic review of ARC's performance by SRI would establish limits, so that the confidence level of SRI management in having their gamble pay off can be reasonable. 3e1

Similarly, ARC needs to be able to adjust its tactics from time to time in order to operate best within the agreed two-year strategic goals and conditions, since there are an uncommon number of uncertainties in our current view, albeit a very high degree of basic promise. 3e2

These latter expectations on our part are basically that we know we have strategic guidelines, but yet neither we nor SRI can freeze all of the tactical factors now. 3e3

Summary of current planning and budget situation: 4

AAG committing to a second TYMSHARE TENEX in January (Office-2, or O-2), then assuming O-3 by 1 Jul 75, O-4 by 1 Jan 76. 4a

O-1, O-3, etc. planned to remain as pure, standard-service systems, generally running only one version of NLS, and not allowing heavy compiling, or other-system usage (e.g. no FORTRAN programs). 4a1

It is evident that continuous, slot-by-slot service expansion isn't possible until we have at least two Utility machines that can each be raised or lowered flexibly in its configuration (therefore, in its cost and its slot-carrying capacity). 4a2

As presently conceived, the "rough-environment" O-2 machine wouldn't have that flexibility. 4a2a

Therefore, there will be like a seven-slot waiting gap between the time that O-1 is filled (at a maximum

configuration, serving 25 slots) and the time we can bring up 0-3 (where 0-1 and 0-3 each would be minimally configured for about 16 users). 4a2b

Note: It is also evident that the old "slot" basis for selling Utility CSO service must be changed to something more nearly an actual "TENEX resource access", where actual machine cycles, storage space, etc, are guaranteed. See below for approach (in discssion of TENEX service). 4a3

Office-2 will be special in Utility's line of computer-service systems. It will be devoted to supporting mixed-system users that are doing extensive development and experimentation. 4b

All of ARC's development work will be done on this machine; generally, 0-2 will otherwise be populated only by other ARC users, or other external people who explicitly are willing and able to live in a "rougher" environment -- e.g. 0-2 would service: 4b1

AKW developers: ARC's ADAG staff, as well as people external to ARC who are developing AKW related things where it is mutually advantageous for them to share our "rough environment." (Early examples of the latter -- MCA, using BCPL to develop the NSW Works Manager system; possibily Norm Nielsen's gang, if they take on development work on the Output Processor.) 4b1a

Applications-support people, to access and learn about forthcoming system features. E.g., AAG staff, perhaps client architects. 4b1b

Selected external clientele using a new system as voluntary participants in its shakedown. 4b1c

AAG's approach, with 0-2, to giving technical-support services, and to charging for CSO services, will be specially tailored for 0-2's special set of system users. 4b2

ARC generally recognizes that it has to provide a special environment such as this. 4b2a

We don't therefore look to this machine to produce the same ratio of ARC income to ARC costs as we will to the standard-service systems. 4b2b

The level of technical service required from AAG to support the special users of 0-2 will initially be estimated at 3 people. It isn't clear yet how much will actually be required,

as chargeable to operating expenses within AAG, For instance, any experimental systems available and used therein would assumedly be the direct business of ADAG to debug and service -- up until the point where they are handed off to AAG for operating and maintaining as part of its regular service. 4b3

ARC is committed to improving its accounting and costing methods towards reaching accurate and realistic cost figures for the services it must provide its various internal activities in support of their work. These activities will further develop their business-charging figures to transfer in some fair way these service costs to client charges and operating overhead. 4b4

AAG planning to establish a graded pyramid of services and associated costs: 4c

Basic TENEX service: This service includes raw TENEX horsepower, operators, file archiving, Journal Ident and Delivery servicing. The users are assumed to be contributing to our AKW goals, but otherwise we will try not to restrict their use of software packages, etc. 4c1

Cost of this: Accounts set up into which all direct charges for these services are accumulated; costs to subscribers then allocated out of these accounts according to percent of resource each client subscribed to. 4c1a

One of the costs here will be associated with keeping running records of resource usage and facility responsiveness, reliability, etc. These must be analyzed, and constant attetion given to keeping high quality service with fair resource access distribution. 4c1a1

An added charge will be levied to this account as a basic "tax" that provides dollars to pay for basic Utility-service developments (a fair amount of this money would go to pay ADAG staff -- but it is AAG's business to collect, specify, contract, an distribute such resource). 4c1a2

Associated condition: To sell raw TENEX service for other than DNLS slots is possible only if there is a practical way to build a resource-usage boundary around a user group. BBN's Pie-Slice scheduler is advertised as being able to do this; AAG will need help from ADAG in getting such a scheduler working appropriately. 4c1b

Possibly, if the scheduler doesn't protect NLS users in the way we'd have to have, there may have to be some

additional practices employed on O-2 in order to let us sell TENEX power in a sensible way -- e.g, people scheduling, or recording actual resource utilization and perodically adjusting the scheduler's percentage parameters for user groups so that what they indeed get on an average corressponds to their fair shares of total sytem resources (in a useable way). 4c1b1

Basic NLS service: This is the standard service for the clients who are doing exploratory application. Usually there is one standard version of NLS being fully supported with documentation, HELP, user training, etc. 4c2

Costing: Separate accounts to be kept for the internal costs of this service, independent of the Basic TENEX services. Some charges are distributed to clients according to the level of service they've contracted for, and others are charged directly to a given client only when certain services are provided directly to him. In any event, the amount established in his service contract will not be exceeded. 4c2a

Special, negotiable services: Special training, installing and maintaining NLS in a client's system, brokering the services of analysis or development people, arranging for services that are better done by AAG than by a client, etc. ("brokering examples": to arrange for communications or terminals). 4c3

Yet to become an appreciable level of business, but exected to grow. 4c3a

Interchange of tangible services between AAG and ADAG 4d

AAG currently agreeing to buy software service from ADAG at average rate of one FTE person begining in December 74. ADAG agrees to have the appropriately skilled help available during that time. ADAG assumes that the person's salary and CSO service costs are provided by AAG (charged against AAG's external revenue, to keep unmixed the overhead costs of running the two different kinds of businesses). 4d1

It is AAG's business to manage this above "ARC CSO" facility, and to negotiate rates. 4d2

Interchange of intangible services between AAG and ADAG 4e

ADAG will be responsible for monitor-development work. AAG will have staff that is equipped to do maintenance work on the

monitors. For instance, in O-2, if the new FE-BE work requires monitor changes, they would be devloped and brought to hand-over quality by ADAG staff. 4e1

ADAG guys are currently exposed to lots of questions by AAG staff. The interchange is generally necessary to bring AAG staff to a stable level of knowledge, and will be specifically important to have go on at some reasonable rate in the future to ensure effective working communication and relationships. For the time being, it is a sort of community-donated and not insignicant cost for which I am appreciative. 4e2

Testing and training of ARC staff: 4e3

AAG will eventually want to see that the knowledge and skill level of every serious AKW user is tracked -- periodically tested, with results explicitly checked at least by the user, expectably by his architect, and if possible by AAG specialists. 4e3a

AAG must also see that the user is periodically made aware of his potential capability increases and of their value to him, as achievable by available training services. 4e3a1

For the time being, we'll assume that some of this testing and training service is offerred free to ADAG (sort of in return for their free question answering). 4e3b

Note: When Testing and Training Services are explicit and well established, I think that every ARC user should be provided them in a definite user-development program, and that their costs be explicitly accounted for by transfer of money. 4e3c

Guidelines for current plan and budget preparation: 5

Growth of application community just about the most basic set of goals. These "dimensions" of growth are relevant: 5a

Size -- reaching how many users, in how many oganizations 5a1

Seasoning of users, of architects, of their management 5a2

Degree of Collaboration -- attitude, perception, practice 5a3

Sophistication (range, depth, ...) of the CSO tools/delivery made available. 5a4

Diversity of applications, or concentrations... 5a5

Basic, longer-term, targeting figures. 5b

For instance, consider that along with the growth in the above "dimensions" of the Bootstrap community, there will be steady increase in the yearly "investment" in analysis and new developments, toward evolving improved system and improved methodologies. 5b1

I'm roughly using a figure of $5 million/yr by the end of CY75. 5b1a

I hope that we can get skilled, professional effort on the market predictions here, for our planning in the future. 5b1b

Development business: 5c

Associated with composite growth rate along various dimensions will be a correlative investment rate in Analysis and New Development. 5c1

Three kinds of business are now apparent as basics to be coveed by the A&D group 5c2

Take on specific development and/or analysis projects within the AKW system. (Assume a groing amount of analysis and development will be done by others, within a framework where central coordination isnecessary). 5c2a

Manage/coordinate all of the "central-system" analysis and development. 5c2b

Application business: 5d

Not plan to deviate very much from the current way the service is sold. 5d1

High priority to CSO delivery quality 5d1a

Tech-support service delivery can be a variable, judiciously chosen. 5d1b

Basic Org-plans and policies: 6

Special-Interest Communities will provide a central orientation to ARC's whol business structure. 6a

Our active new-business promotion will be guided by: 6a1

Enlistment of new clients to join specifically designated Special-Interest AKW Communities, we won't "actively pursue" a client otherwise. 6a1a

"Walk in" prospective clients will be told that it is only by joining an existing Community can we consider selling them Utiliy service. 6a1b

If we see extraordinary value to our Bootstrap Community growth in taking on some new client where no appropriated SI Community has been designated, then we will give him special consideration: 6a1c

We may charge him more for Utility service, because he will cost us more to bring him along. 6a1c1

We may designate a new SI Community into which he fits, if this is deemed sensible in light of the whole BC evolution. 6a1c2

We may, indeed, explicitly "not offer to sell him our services," bacause it doesn't fit out plans and/or current situation. 6a1c3

Community-nucleation activity will be coordinated for the time being directly by DCE. Direct service negotiations and contracts assumedly fall out for ADG and AAG, which specific contracts are then to be handled by Dick and Jim. Other arrangemets expected to be made later for central coordination here. Crrent community nucleators are: 6a2

Nucleators and communities already arranged for: 6a3

Jean Iseli -- NSA, possibly others in Intelligence Community (Jean is a MITRE employee; has substantial involvement now with NSA in their conversion to ARPANET-like internal computer-utilization system.) 6a3a

Phil Whalen -- DCA, possibly others in Intelligence Community (Phil is Director of Systems EValuation Department, in SRI's Engineering Systems Division) 6a3b

Dirk Van Nouhuys -- DPCS (Dirk will become a member of Norm Nilsen's Information Systems Group, in Dave Brown's Information Science Laboratory. 6a3c

Ernest Anastasio -- CBI (some uncertainty here, until he and Col. Kibler of ARPA HRRO reach agreement. Kibler had thought that O'Sullivan ad had a man at ISI pegged for this

role, Anastasio is with Educational Testing Services, Princeton, N.J.) 6a3d

Tentatively planned-for Communities
(Note: "(xxx)" signifies "Nucleator as yet undetermined") 6a4

(xxx) -- SEAS (to be actively launched this Winter) 6a4a

(xxx) -- General Community Service Center Community (a SI Community whose special interest is in serving a distributed community, as per -- 12445,) The NIC is an existing case, as will be (perhaps) the ETS group for the CBI Community, and potentially some group in each of the SI Communities; also, some new clients will come on essentially just to develop that particlar role for some community. 6a4b

In fact, many SI Communities can potentially be formed, by outside nucleators, whose special interest is not in an area that has bootstrapping leverage for AKW Systems. Our initial support for them could (by policy on our part, perhaps) require that we only deal with their community nucleator and their community "Workshop Service Center" (and perhaps the funding agencies who will support all this). 6a4b1

(xxx) -- Computer-Aided Design Community. (Note: likely that the most sensible starting place for such clients anyway would be for DPCS operatons, especilly integratin text and graphics. Before we explicitly launch a CAD Community, we'd actively develop a strong subset of DPCS Community participants who will make a good nucleaus for graduating into more extensive CAD AKW work warranting a SI Community for CAD operations outside of their DPCS work.) 6a4c

(xxx) -- Decision Analysis Community (ARPA's HRRO appears to be assumbling such a community, more or less patterned after their CBI Community.) 6a4d

(xxx) -- Large-Project Management Community (Potentially, the SDMS Design Center fits here. A very important specialty area, with lots of (sensible) overlap with the sub-speciaties of CAD and SEAS.) 6a4e

(xxx) -- ICIAS Community (Information Collection, Integration, and Analysis System). The basic, functional operations of an ICIA System are central to almost every augmented workshop, and every Utility client would benefit from what can be accellerated here. But for some organizations, thse functions already have enough central

importance to warrant their being very active in an SI Community aimed at improving this set of "Kernel operations". 6a4f

SEAS (Software Engineering Augmentation System) will be pursued actively as a community development process. 6a5

The ARPA/AF NSW Program forms an extrmely powerful basic impetus to help get a full-fledged SEAS Community going; hopefully, NSW and SEAS would remain in effective collaboration. 6a5a

The NSW Program, as it is currently perceived by Carlson, can only accept DoD agencies as legal, contractural participants. There is no provision for enlisting collaborative participation from non-DoD government agencies, nor from non-government outfits. 6a5a1

I explicitly asked Carlson if he saw any problems in our going ahead with a SEAS plan that could provide de-facto collaboration; he explicitly said that he foresaw no problem, and had no personal objections (could see the potential advantage to all concerned). 6a5a2

Basic picture of how we can approach the SEAS Community: 6a5b

We count on our AKW Utility developing the capability to arrange for access, by non-DoD clients, to: 6a5b1

a version of the NSW Works Manager, 6a5b1a

any of the importat tool systems from the NSW world. 6a5b1b

We go ahead with a SEAS Community Nucleation plan pretty much as we would for any special-interest community, except that we assume: 6a5b2

The NSW Program has first option on any DoD clients -- in fact, we'd sort of treat the NSW Program manager as though he had "our SEAS Community Nucleator franchise" for DoD agencies. 6a5b2a

For initial, practical purposes, we consider that our SEAS Community will consist only of non-DoD organizations -- the Nucleator's franchise will explicity be limited to such. 6a5b2b

There is obvious advantage in the possible "happy relationship" downstrea where practically speaking tere

is close collaboration between the union of NSW and SEAS participants, and between their repsective nucleator/manager agents. Let's assume that this will come about, and that all parties would not only benefit significantly but would perceive the advantages. 6a5b3

In the early years of the SEAS Community, ARC should be able to promote many activities that it can perceive as directly complementary to NSW's. As The xx matures, it would be expected that we still can communicate such perceptions, but that the Community would more and more develop a seasoned mind of its own. 6a5b4

SEAS-community nucleation won't be in ADG, as had previously been contemplated. Instead, it will be somebody outside of ARC. 6a5c

ARC's Development and Analysis Group (ADAG) will cover the following basic roles: 6b

Coordination of all development and analysis work, on "The AKW System", done by any group (ARC, other SRI, outside SRI). This involves the "Coherent AKW System", for which it is ARC's central purpose over the next few years to promote coherent evolution and availability to bootstrap-contributing applictions. 6b1

Here, for instance, stems responsibility for seeing that the needs are clearly spelled out for D&A work that is important to System evolution and delivery. Also, for seeing that Analysis activity gets established in methodology, impact, and recognition of its value. 6b1a

Specific Development work on The AKW System 6b2

Specific Analysis on the AKW System 6b3

ARC's Applications Group (AAG) will cover the following activities: 6c

Plan that its types of CSD-delivery service steadily expand -- e.g. as below: 6c1

NLS in Utility TENEX (as now) 6c1a

Install and maintain NLS in client's own TENEX 6c1b

Support NLS terminals for remote clients. E.g.: 6c1c

DNLS/LINE-Processor and steady stream of appropriate displays 6c1c1

Graphic manipulation terminals (as evolved by NSW, and further under DPCS impetus). 6c1c2

Support Frontend (in ELF) -- sometime between Jul 75 and Jan 76. Frontend at Utility site, either in an 11 or in TENEX. Aso, support Frontnd software in Utility Client's site. Assume that Frontend software can be debugged and mantained fully from ARC. 6c1d

Broker/provide NLS Frontend-Backend service where Backend CSO service obtained via one or more commercial timesharing companies, available in manner usually arranged (e.g. set up account, billed for resources used, etc.) 6c1e

Install and maintain NLS in client's own Frontend-ELF and non-TENEX Backend computer. 6c1f

Broker/contract for providing alternative/improved digital communication to the NLS Utility. 6c1g

Plan that it's technical-support services expand -- e.g.: 6c2

Testing to establish levels of skills and knowledge possessed by a given knowledge worker in relevant areas of knowledge work. (Towards fully-automatic computerized testing, self administered.) 6c2a

Specialied training courses, for raising skill and/or knowledge up to prescribed levels in a given knowledge-work domain. (Toward fully automatic computer-based instruction.) 6c2b

E.g., for the group that will run the Documentation Development, Production and Control System for an organization. Or, for a team of software developers that is moving into its augmented workshop. Or, for L10 user-programmers to support the user-program needs of their local users. Or, small-team, boss-secretary groups that are going to work together in their augmented office. 6c2b1

Operate the centralized part of an AKW-System Research Intelligence System to support developers, analyzers, planners, users out in the clientele community. 6c2c

Operate/broker complete services for typography

consultation-design and photo-typesetting of clients' publications. 6c2d

For meetings and conferences that client groups want to hold in a highly augmented environment, handle local arrangements, provide for special spaces, equipment, and support by specialists in "augmented meetings." 6c2e

This can be for a "private working group, among people already involved in AKW usage; or, it could be to support a working meeting tat incudes a good number of people unskilled in AKW techniques; or it could include supporting an AKW-seasoned group that wants to put on a presentation to a large group in the manner we did for FJCC in 68 and ASIS in 69. 6c2e1

Plan that the flexibility of its negotiations, contracting, service charging, etc, steadily improve. 6c3

Working relationships between ADAG and AAG: 6d

Regular flow of contract business between them, with mutually satisfactory formalization, bookeeping, etc. 6d1

Regarding "analysis" work: Applicaions is running a complex business. Planning its future requires extensive analysis -- some of which involves the AKW System performance, delivery characteristics, etc, which is the responsibility of ADAG to coordinate. Some of the analysis in AAG's Plas & Analysis activity concerns business operations not within the AKW-System purview (i.e. instad, concerning the brokering, charging schedules, impact on usage patterns of given cost schedules, etc.). I'm not clear how we should practically deal with the latter kind of analysis; there is some reasonability about either group as a home for it. 6d2

Without prejudicing later settlement as to where "home" will/should be, I'll put the AKW Sytems analysis under ADAG, and the business-operations analysis under AAG. Being "under" says whether it is Dick or Jim who is responsible for providing the staffing, methodology, standards etc. needed to support that kind of work. 6d2a

Regardless of who provides that "home", the future contractors for any particular kind of service have to anticipate their requiremnts, and contribute honestly to planning and budgeting exercises toward ensuring that they can have the necessary services available. 6d2b

Also, as far as I am concerned, the same analysis guy(s) can participate in either kind of analysis activity under this specified "home-base" arrangement. We have plenty of need anyway for our task-workers to keep track of different people who may have different review-responsibility roles with respect to a given task. 6d2c

Regarding "software" work. Quality assurance for Utility-supported software is clearly for AAG's software staff. A certain amount of tinkering with the top parts of user programs also. But any new twists to function, practice, etc. in software belong under the "devlopment-coordination" domain of ADAG -- I'd guess that within ARC, any sizeable software job would be done by ADAG staff under contract to someone. ADAG can, if mutually agreeable, give explicit clearance to AAG's software staff to do any development task (under similar control/coordination measures as if being done by any group outside of ADAG). 6d3

Regarding "Quality Control" and other new-product characteristics that affect the valid business of AAG. 6d4

Whenever a new thing is being developed by ADAG, it will generally be expected that it is destined for operational support later in the AKW Utility. If such is not the case, then that particular ADAG effort would be viewed as a "loss" -- which may for some reason still represent good choices over-all. 6d4a

Sometimes a development item is directly aimed toward integration into the Utility, in which case AAG from the outset needs to be involved in reviewing requirements, specs, implementation approach, anticipated side-effects, etc. 6d4b

Sometimes a development item is aimed toward satisfying a specific need of a customer that isn't planning on its later application being within ARC's "standard" Utility environment. 6d4c

NSW comes close here. Although the NSW Program people may later elect this option, they so far haven't committed themselves as to the service environment or NLS version they want to have NSW user service provided.) 6d4c1

We assume that the basic characteristics of the development lie so closely to what is desireable for Utility servive evolution that we will take the contract, do a good job for the sponsor, and count on potetially

doing a retrofit to the Utility's needs later, still to good over-all gain. 6d4c2

In such a case, AAG doesn't have as clear a position with respect to reviewing and affecting relevant features of the development project. We need to work on this sort of case. 6d4c3

For cases where the sponsor definitely isn't aiming for his implemented thing to be used in an AKW System maintained by our Utility: 6d4d

I'd say that at least AAG should expect normally to be able to review the ADAG project, unless there is direct and valid reason otherwise. Similarly, I think AAG should be able to offer its recommendations, and I'd expect ADAG to give fair consideration (if it isn't to the sponsor's detriment, consider acceding to AAG suggestion). 6d4d1

The cost of any slow-down caused here by ADAG's supporting AAG here must definitely not be charged to the sponsor, and basically must come out of the business costs for AAG's studying and arranging for new services to come up on the Utility. 6d4d2

A fully cooperative attitude on ABAG's part here (when possible within existing pressures), would find them giving reasonable consideration to doing the following: 6d4d3

Suppose AAG sees where a bit of extended effort now would return much benefit in their world later 6d4d3a

ADAG would do an add-on study, paid for by AAG, to check out implementation costs, affects in the sponsor's world, etc., and work therin toward specifying an approach that would also provide extra benefit to the sponsor. 6d4d3b

Then ADAG would bring up the 6d4d3c

Bootstrapping within ARC 6e

For years ARC operated under the policy that it would bootstrap the AKW System, using what it developed, and developing only what would make its knowledge work more effective. 6e1

Since the time we became involved with the NIC, the

difference between this principle and our practice has steadily widened. As of 1 Jul 74 it essentially has been abandoned. 6e1a

What is operative now is pretty much that use of our AKW tools has to be paid for out of contract or overhead expenses as can be justfied. There is no external support being provided for ARC to experiment -- whih I know is most ironical and which I intend to see changed (ironical in that we are servicing other groups that are being paid for experimenting). 6e1b

I don't believe that it is absolutely necessary, for achieving near-future goals, that ARC does get extra support expressely for the extra rsources (computer, terminals, internal workshop architects, etc) to effectively pursue evolution of its own augmetation. 6e2

But I can't imagine how ARC can carry out its longer-term role effectively if we don't within a year or two get active again about our own augmentation. I'd like us to be the best example in the world of a group workshop for Information-Sytem Development, for Information-System Analysis, for Complex-Project Management. 6e3

I have a number of alternative approaches in mind, toward improving this situation. Perhaps the strongest of them is based upon the success of the SI Communities, and the case that different activities within ARC can make for spearheading application of tools and techniques relevant to particular Community (and to its participating members and interested sponsors). 6e4

More on his topic later ... 6e5

DCE 27-FEB-75 18:12 25507

ARC Planning Notes from 3 Nov 74

(J25507) 27-FEB-75 18:12;;;; Title: Author(s): Douglas C. Engelbart/DCE; Distribution: /SRI-ARC([INFO-ONLY]) ; Sub-Collections: SRI-ARC; Clerk: DCE;

moving nsw files

Note: 1

This needs to be combined with the file structure types and the conversion procedure ideas. 1a

Introduction 2

This is a description of the procedures involved in moving a NSW file from one file package controlled location to another file package controlled location. 2a

General Structure 3

filespec = LIST (directory, password, filename, element-spec) 3a

fileelm = LIST (ph, did, filename, element-spec) 3b

COpY (NSW => NSW) 3c

Internal Works Manager routine looks up the tWo file references and determines the source and destination locations, and source and destination file package names for the files. 3c1

Internal Works Manager routine creates a channel between the source and destination file packages (which are already open) by calling on the local process management package. 3c2

CRTPHYCHN (ph1, ph2 => poh1, poh2, pcn) 3c3

The internal Works Manager routine calls the PULLFILE procedure in the "macro file package" at the source location. 3c4

PULLFILE (filespec, dstype, dst => value) 3c5

This routine simply parses out the filespec arguments and calls the file package to first open the directory and then get the file. 3c5a

OPNDIR (directory, password => did) 3c5b

GETFIL (fileelm, disp, dst, dstype => value) 3c5c

The file access parameters are checked and then the sub procedure getit is called. 3c5c1

GETIT (fileelm, disp, dst, dstype => value) 3c5c2

This routine actually reads the file from the local file system and send the file via the IPC procedure

SNDMSG, generally this will require a series of file reads and sndmsgs. 3c5c2a

SNDMSG (pohi, message) 3c5c2b

The internal Works Manager routine calls the PUSHFILE procedure in the "macro file package" at the destination location. 3c6

PUSHFILE (filespec, srctype, src) 3c7

This routine simply parses out the filespec arguments and calls the file package to first open the directory and then put the file. 3c7a

OPNDIR (directory, password => did) 3c7b

PUTFIL (fileelm, disp, src, srctype) 3c7c

The file access parameters are checked and then the sub procedure putit is called. 3c7c1

PUTIT (fileelm, disp, src, srctype) 3c7c2

This routine actually receives the file via the IPC procedure RCVMSG and stores the file to the local file system, generally this will require a series of rcvmsgs and file stores. 3c7c2a

RCVMSG (pohi, message) 3c7c2b

Particular Structure 4

NSW-file = LIST (filespec, locspec) 4a

filespec = LIST (directory, password, filename, element-spec) 4b

fileelm = LIST (ph, did, filename, element-spec) 4c

Note: the notation 4d

listname(i) 4d1

indicates the "i-th" element of the list "listname". 4e

COPY (NSW-file => NSW-file) 4f

CRTPHYCHN (ph1, ph2 => sport, dport, pcn) 4f1

PULLFILE (filespec, CHNL, sport => value) 4f2

```
        directory <= filespec(1)                                    4f2a
        password <= filespec(2)                                     4f2b
        OPNDIR ( directory, password => did )                       4f2c
        fileelm <= LIST ( did, filespec(3), filespec(4) )           4f2d
        GETFIL ( fileelm, RETAIN, sport, CHNL => value )            4f2e
           GETIT ( fileelm, RETAIN, sport, CHNL => value )          4f2e1
              loop:                                                 4f2e1a
                 message <= % a portion of the file element
                 specified encoded as          a PCP data
                 structure %                                        4f2e1a1
                 SNDMSG ( poh1, message )                           4f2e1a2
                 if not end of file then go to loop                 4f2e1a3
              end                                                   4f2e1b
     PUSHFILE ( filespec, CHNL, dport )                             4f3
        directory <= filespec(1)                                    4f3a
        password <= filespec(2)                                     4f3b
        OPNDIR ( directory, password => did )                       4f3c
        fileelm <= LIST ( did, filespec(3), filespec(4) )           4f3d
        PUTFIL ( fileelm, RETAIN, dport, CHNL )                     4f3e
           PUTIT ( fileelm, RETAIN, dport, CHNL )                   4f3e1
              loop:                                                 4f3e1a
                 RCVMSG ( poh1, message )                           4f3e1a1
                 % a portion of the file element specified decoded
                 from          a PCP data structure % <= message    4f3e1a2
                 if not end of file then go to loop                 4f3e1a3
              end                                                   4f3e1b
```

moving nsw files

(J25508) 27-FEB-75 18:39;;;; Title: Author(s): Jonathan B. Postel/JBP; Distribution: /JEW([INFO-ONLY]) DSM([INFO-ONLY]) ; Sub-Collections: SRI-ARC; Clerk: JBP; Origin: < POSTEL, FILE-COPIES,NLS;4, >, 27-FEB-75 18:35 JBP ;;;;####;

DCE 27-FEB-75 18:42 25509

Residue from preparing NSF-OSIS SSSC proposal (24758,)

See also the OSIS RFP (XDOC -- 24354,)

DCE 27-FEB-75 18:42 25509

Residue from preparing NSF-OSIS SSSC proposal (24758,)

New grist: for ... 2E INSTITUTIONAL RESOURCES AND RELATED PROGRAMS 1

In SRI's Telecommunications Sciences Center, there is a project headed by Roger Hough, entitled "Teleconferencing Systems: A State of the Art Survey and Preliminary Analysis", being done for the National Science Foundation under the "Social Systems and Huma Resources" Program. The project is surveying sites and systems that have been established in the past. Some video systems go back to 1960. 1a

Hough Notes: 1b

"Teleconferencing" in their terminology is group communications -- so common thing is to have conference rooms at dispersed points, connected by video etc. 1b1

Most widespread connotation is electronic mediated group communications, groups in real time, having some or all o thir communications be via electronic media. 1b1a

Or alternativelY, several persons, doing confernce ... 1b1b

Mostly real time 1b1b1

Many more examples of oice and video specific activities than computer conferencing -- which is by ffar the smallest segment. 1b1b2

more than two people... otherwise essentially "Distributed Dialogue in 'conference mode'" in AKW language. 1b1b3

How would their df rule out professiona journals... Because they talk of electroni communications (telegrams would be in... if broadcast) 1b1c

Identifid 40 or 50 sites/sYstems, either operational or experimental. Most of them experimental. Only about 5 are computer-communication. Rest are split about half/half in having video or not to supplmet the audio. Som have facsimile-graphics supplement. 1b1d

Computerized is our only comparable.. 1b2

FORUM 1b2a

Turoff 1b2b

Bell Northern reserach doing spinoff 1b2b1

DCE 27-FEB-75 18:42 25509

Residue from preparing NSF-OSIS SSSC proposal (24758,)

OEP leftover 1b2b2

Cite (From Vallee's paper, p 123 1b2c

Turoff 73, 1b2c1

Vallee 74 1b2c2

Conrath, 72 1b2c3

Schuyler and Johansen, 72 1b2c4

Lots of general lit on such as subst communic for travel, etc. 1b3

Any prior Teleconf stuff been used like for our Group Study application? 1b4

He doesn't know. 1b4a

Turoff's aim sort of to support policy-setting etc, deliberation on a fairly urgent topic. Still being used ".. in crises situations.." 1b4b

Old Grist: 2

Residue from preparing NSF-OSIS SSSC proposal (24758,)

Grist from Draft1/2 2a

This draft is aimed at Category 2 of the Program Solicitation NSF 74-38 (ARC XDOC -- 24354,), entitled "Improved Disseminaton and Use of Scientific and Technical Information." 2a1

Support center for special study efforts by disbributed groups. 2a2

Aimed to support the participants of a specially commissioned study of some topic, where: 2a2a

reference information is generally to be gathered, 2a2a1

a framework for a report is developed (evolutionary, perhaps), 2a2a2

objectives clarified, 2a2a3

terminology settled, 2a2a4

main issues isolated, 2a2a5

sub-groups assigned special study areas, 2a2a6

memoranda exchanged, 2a2a7

trial drafts and thinkpieces produced and commented upon and reviewed etc, and then revised, 2a2a8

formal bibliographic accumulation, etc,,. 2a2a9

Essentially all of the basics are available now to support this, Experience of DCE in the two-year INFOSYS panel was relevant; note the bibliographic support provided then, as well as rudimenatary draft-cycle support. 2a2b

Request support for ARC staff to: 2a2c

solicit, in conjunction with OSIS staff, candidate study groups, to whom this service would be offered on a special-cost basis (cOst issue discussed below -- basic assumption is that OSIS wouldn't have to bear all of the operaional cost of a study-group's support). 2a2c1

Help design the procedures, methods, information structuring, etc, for the particular group, to suit their particular nature. 2a2c2

Residue from preparing NSF-OSIS SSSC proposal (24758,)

Provide constant monitoring and consultation; towards helping them around troubles, developing more value, etc. 2a2c3

Assume that Utility capacity is procured for them, but that the extra-heavy distribution, plus the transient nature of the participants, makes their ratio of help need to their Utility charges too high to have all of their help come from normal Utility charges. 2a2c3a

Develop special additions/extensions to the Utility service thatimprove its effectivess and/or reduce its operational cost for this particular kind of applicaton. E.g.: 2a2c4

Flexible set of batch-spooling and deferred-execution support services (OCR, paper tape, spooling on local hosts or specially set-up minis, spooling in commercial T-S sYstemer, dial-up modem teleprinter, microfiche, ... 2a2c4a

Complete microfiche services for both compter-held material and XDOC, including computer-supported cataloging, indexing, retrieval, and access operatons. 2a2c4b

Special documentation, training, computer-aided instruction techniques that serve the isolated participant (and especially his secretarial helper). 2a2c4c

Distribution-control processes, for keeping track of drafts, reviewers, authors, and their due dates, their sign-off protocol, etc. 2a2c4d

Coordination with the chosen EPC for production of the final report(s). 2a2c4e

Category-2 RFP Spec: (osisrfp,2:gebbtzn) (osisrfp,1:xhzb) 2a3

Category 2: Innovations and improvements in science communications systems and services 2a3a

Emphasis is upon 2a3a1

information "intermediaries" and services 2a3a1a

which provide 2a3a1b

condensation, 2a3a1b1

filtration,	2a3a1b2
validation and	2a3a1b3
relevance functions.	2a3a1b4
Innovations in **	2a3a2
publishing,	2a3a2a
retrieval,	2a3a2b
and user services **	2a3a2c
are eligible.	2a3a3
Projects may be	2a3a4
design and development efforts oriented to **	2a3a4a
specific innovation	2a3a4a1
in the	2a3a4a1a
technology,	2a3a4a1a1
organization and	2a3a4a1a2
management	2a3a4a1a3
of scientific communications,	2a3a4a1b
or operational experiments designed to **	2a3a4b
test and validate	2a3a4b1
hypotheses	2a3a4b1a
or promising innovations	2a3a4b1b
developed from prior research efforts.	2a3a4b2
Areas of special interest include:	2a3a5
improved retrieval techniques and capabilities,	2a3a5a
innovative products and services **	2a3a5b
which can be offered by	2a3a5b1

libraries and 2a3a5b1a

information service centers, 2a3a5b1b

cost-effective 2a3a5c

information storage media and 2a3a5c1

retrieval/delivery technology, 2a3a5c2

paperless publication by means of micrographic and electronic media, 2a3a5d

cooperative resource sharing, 2a3a5e

cooperative endeavors in document delivery services, and 2a3a5f

user-oriented publications, 2a3a5g

Innovations and improvements may be 2a3a6

in the 2a3a6a

technology, 2a3a6a1

organization, or able 2a3a6a2

management 2a3a6a3

of 2a3a6b

scientific communication and 2a3a6b1

technology transfer 2a3a6b2

services, 2a3a6c

ARC/BC/DCE potential for each category: 2b

Category 1: Improvements in the managemnt and use of scientific and technical information in industrial settings 2b1

"Systematic studies are required ..." -- direct impliation that only study-type of projects, 2b1a

Category 2: Innovations and improvements in science communications systems and services 2b2

Open for most of DCE/BC basic explororatory development and application: e.g. RINS/HNDBK kickoff 2b2a

RINS/HANDBK support for a number of the projects here could be considered; at least, the Utility services would be available, and ARC-P2 (ARC Project under Category 2) could work on the conventions, techniques, liaison, etc. 2b2b

Dialogue among all of the different participants could be most beneficial. 2b2c

Consider some simplest way to support Journal/XDOC dialogue between OSIS and its contractors here. 2b2c1

Minimal terminals required -- TTY33 dialup; TWX; mail; phone, etc. 2b2c2

Each contractor like one of the old NIC stations. 2b2c3

Consider proposing ufiche for ... DDSS, RINS, HNDBK... 2b2c4

Note the running start on RINS/XDOC collections via NAS/INFOSYS, NIC, our general XDOC. 2b2d

Seemingly special relevance for RINS/DSS/HNBK support -- in Project 2b2e

Category 3: Tests and analyses of marketing strategies and techniques 2b3

Consider marketing in "cooperative sys dev and service support" domain, making it as directly useful in BC world as possible. 2b3a

Would need to be other parties do the study here, people skilled at that kind of work. 2b3b

Category 4: Economic characteristics of scientific and technical information communication 2b4

Cf 4c: Data base -- STI RINS correlation? 2b4a

Again, perhaps direct tie to the BC world, its economic characteristics? 2b4b

Would need to be other parties do the study here, people skilled at that kind of work. 2b4c

Category 5: Performance evalution of services and systems 2b5

Special features in BC world? Evaluate services and systems, as for the STI world of tomorrow? 2b5a

Would need to be other parties do the study here, people skilled at that kind of work. 2b5b

Category 6: Energy R&D information 2b6

The old "evolutionary, community info-service center, etc." -- how about a design plan, consistentwith the general bootstrapping approach in the other AKW and STI things that ARC relates to? 2b6a

Category 7: Data tagging -- design studies and operationa experiments 2b7

BC world needs to work with data and analysis. Many of its prime clients would, too. Would be neat to get a leg up. 2b7a

Could one do the sort of planning and designing, perhaps a bit of prototypical experimntation, under OSIS support, to where it would really show one of the larger-funded data-analysis clients how they could go? Check with Dave Maynard. 2b7b

SRI's large-scale data-analysis programs could be buyers; NSA also; probably many. 2b7b1

Could help a lot if ARC could get support to carry through the formulation and convincing stages. 2b7b2

Category 8: Editorial processing center -- operational experiments 2b8

A flaming natural, this one. 2b8a

Current plan: 2b8b

SRI (via Nielsen's group, involving Placko and DVN undoubtedly) propose the following: 2b8b1

Plan to set up an operational, prototype EPC, to do the whold job on editorial processing for technical reports and journal papers. 2b8b2

It would be based upon AKW Utility tools -- only usng what is available to any Utility Client in the way of comptuter services. 2b8b3

Probably involve new features being developed; but expect largest prt of support to go for the procedures, testing, evaluating, etc. 2b8b4

Would run editorial-process jobs for people at some reasonable cost (e.g. at first being comparable to old ways; OSIS project taking up the slack. When get thehang ofit, and really know how to do it and what the break-even cost would be, see if can get business charging that. If not, determine where best could invest in order to do so. When et he service shaken down, offer it to selected (like with mutual selection with OSIS staff) clients on the outside -- remote service.) 2b8b5

Category 9: Assessment of the health of the scientific and technical communication system of the United States 2b9

General: possible launchings: 2c

Consider starting with just some of the OSIS staff (initially, just the TIP Program staff) as a first nucleus of users, at their Penn-Ave offices. 2c1

Later, more of the OSIS staff, and organizational units outside OSIS, even outside NSF. 2c1a

Would somebody have the time required of a KW Architect role? 2c1b

Characteristics of the KW Architect: 2c2

A person accepted in the user community. Needs god personality; should be young enough to devote time and energy heavily toward larning the skills, gaining the specific experience in training, in exploring the new orking methjods, etc. 2c2a

What needs to be done, for BC: 2d

Provide an added attraction 2d1

to key types of clientele 2d1a

in the way of 2d1b

relevant inelligence and handbook stuff 2d1c

Establish prototype 2d2

of such nuclear, community information services 2d2a

for adoption by other-oriented communities 2d2b

especially a subsequent, formally pursued STI Community 2d2b1

Provide for coherent extrapolations of NP 2d3

that are kept visible to all and 2d3a

that all relevant parties participate in 2d3b

Provide special base for complex-system developers 2d4

especially information-system developers 2d4a

Get development going for support to distributed community participants 2d5

e.g., services that support special-collection hard-copy 2d5a

like ufiche support in the RINS, DSS, Handbook domains 2d5a1

Trial proposal spec, under Category 2: 2e

Towards core for BC's nuclear info services 2e1

Operative support for developers of complex information systems: 2e1a

Already there are services that basically support DSS 2e1a1

Know that improvements are in line. Could either propose implementing some of them here, or just gauranteeing to provide the "NP collecting, coherent presentation, etc." support so that others can buy and develop. 2e1a1a

The DDPCS (EPC in their terms) part of this is being adressed under Category 8 -- even if not done with SRI's approach ter, the community tht this proposed project works in will have DDPCS support, and a commitment towards building a special-interest community that will be developing it.. 2e1a2

RINS has a basic mechanism (XDOC catalogs, Journal, etc.), needs a specific application effort, to coalesce methods, proviide clear examples, that can support the collection 2e1a3

Basic tools for Handbook development and publication alreay exist, and for control of its distribution, and for "Query" ype of on-line access. 2e1a4

ARC Locator is a prototype, covering the basic CSO part of our AKW. 2e1a4a

Towards an AKW System-developer's Handbook, need work on 2e1a4b

terminology, 2e1a4b1

completeness in depth 2e1a4b2

implementers as well as users, 2e1a4b2a

user org&method as well as features, 2e1a4b2b

cross referencing 2e1a4b3

Yet to be done is putting it together as broader-scope thing 2e1a4c

THE CBI part has HELP as a tentativ start, 2e1a5

References: 2f

(bprox, pcitnum:wzg) for Citation-renumbering program 2f1

(bprox, plsP:wzg) for lead-SP forcing of all statements 2f2

["(App" 1$D ')]; 2f3

In January of 1974, ARC began operating an "AKW Utility Service" as described in (Apx11), (App-11), (P6Draft, App-1), (Ref3), (Ref3), Basically, it provides: 2f4

(App-11), ddddddd 2f5

(proposis,1:hctKi) 2f6

SINCE (11-DEC-74 15:55); 2f7

SINCE (12-DEC-74 01:55); 2f8

(prop) (proposis,1:xhbznA) 2f9

(App-1) Appendix 1 3

Residue from preparing NSF-OSIS SSSC proposal (24758,)

[NOTE: Plan for this to be built up into a reasonably detailed, coheent description of the tools and techniques applicable to the proposed service. Originally developed this text under "Institutional Resources ..." Section (2E) -- all of the "(App-)" references in this text, and the propsal body, will have to be bumped by 1 to enable this Appendix 1 t be inserted.] 3a

NIC Notes (nnnn,) 3a1

Shaky hard years; learning; 3a1a

Difficulty with large clientele of low training level -- would expect Special Study Groups to have low-trainables, but not so many, nor so unreachable, nor uncontrollable. 3a1b

Particular service features of note: 3b

CML 3b1

HELP and ? 3b2

Line Processor 3b3

Telenet 3b4

Calculator 3b5

DDPCS Community Plan 3b6

NSW impact; 3b7

Little things that can help: 3c

Citation control during development of a document 3c1

User programs for message management 3c2

Correspondence management using Journal and XDOC 3c3

Bibliographic control for distributed collaborators 3c4

Private-collection management, mixed Journal and XDOC 3c5

Mini-essays 3d

Microform collections, indexing, retrieval, and access 3d1

Spectrum of textual communication media 3d2

Residue from preparing NSF-OSIS SSSC proposal (24758,)

< ENGELBART, C2NOT,NLS;2, >, 14-DEC-74 18:31 DCE ;;;; Title: 4

DCE 27-FEB-75 18:42 25509

Setup and Operation of Prototype, Computerized, Research-Intelligence and Handbook Services for a Technical Discipline

ABSTRACT 4a

A small, specialized, scientific-technical community, pursuing a common discipline or mission, can benefit from special information services that are the trial prototypes of future, generally available STI systems. 4a1

This proposal is for setting up and operating a core, prototype, "research-intelligence and handbook" service center. 4a2

The topical area covered would be that of knowledge-workshop systems. 4a2a

The center would use the basic computer tools available operationally from SRI's AKW Utility (see Appendix A). 4a2b

The center would initially serve a relatively small community under OSIS funds; the techniques, methodology, information structuring, etc. developed would be capable of supporting service in other larger centers that later can serve other scientific-technical communities. 4a2c

It is expected that several existing, on-line, special-interest communities, and a number of subsequent such, will support the operational expansion of this center, towards its becomin a ull-blown prototype serving widely distributed, dynamic community. 4a2c1

The project would support the person carrying the chief-architect and editor-in-chief roles 4a3

and in his setting up te core methods, information bases, knoledge structui g, and specially adapted user features. 4a3a

** 171 4a4

** 0 4a5

NARRATIVE 4b

Introduction 4b1

Project Goals and Detailed Objectives: 4b1a

Goal 1: To make operational a small, special-purpose "research intelligence" system (RINS), comprised of: 4b1a1

A special collection of working-grist information:

making up the relevant material to support the "chief architect" of a long-term, very-large, systems project. This "intelligence base" exists in two basic forms: 4b1a1a

External-document (XDOC) reference -- books, reprints, memos, clippings, equipment catalogs, photographs, movie film, photographic slides, letters, etc. 4b1a1a1

Computer-held text, comprising notes, memoranda, reports, etc. within a computerized, full-text memoranda and publication system (the NLS Journal). 4b1a1a2

A computerized catalog (see Appendix A) for both the XDOC and the Journal collections -- 4b1a1b

from which hard-copy indices, shelf-listings, etc. can be produced for special sub-collections; 4b1a1b1

to which there is on-line retrieval mechanisms; 4b1a1b2

to which there may be on-line citation linking from computer-held text (e.g. from NLS Journal items -- see Appendix B). 4b1a1b3

Study Plan 4b2

Organization and Management Plan. 4b3

Dissemination of Results 4b4

Institutional Resources and Related Programs 4b5

Personnel 4b6

BUDGET 4c

(q) Questions to be answered: 4d

How many XDOC entries? Journal entries? (c2not,023) 4d1

```
< ENGELBART, OSISRFP.NLS;10, >, 2-DEC-74 11:45 JML ;;;; ,DIR=1;
,DLD='@; @HED=
"Structured notes from OSIS/RFP (XDOC -- 24354,) @SPLIT; DCE
23-NOV-74";
TC( LevClip=2; Trun=1; SCR=1; @PLEV=1; / SCR=2;)
    SINCE  ( 16-OCT-72 3:20);
@MCH=65; @SNF=72; @DLS=0; @PGN=0; @PES;  @YBS=1;
@HRM=72; @F=
"[Their exact words, segmented and structured by DCE] @SPLIT;Page
@GPN;"; @FRM=72;
```

5

Category 1: Improvements in the management and use of scientific and technical information in industrial settings 5a

Engineers and R&D scientists working in industry comprise a very large and highly scattered user group, and are generally not well served by present information services. 5a1

Despite the size of this group, their information needs and value returned by services to them have not been adequately studied. 5a1a

Systematic investigation is required before major improvements in services to industry are developed. 5a2

Examples of topics which need to be addressed include: 5a3

the relationship between R&D productivity and availability of or form of STI delivered; 5a3a

roles of corporate library or information service centers in helping users; 5a3b

information about purchasing patterns and behavior (institutional vs. individual); 5a3c

analyses of the motivations of information users; 5a3d

effects of management policy and attitudes on technical information utilization; 5a3e

and barriers to effective use of scientific and technical information in industrial settings. 5a3f

Category 2: Innovations and improvements in science communications systems and services 5b

Emphasis is upon 5b1

information "intermediaries" and services 5b1a

which provide 5b1b

condensation, 5b1b1

filtration, 5b1b2

validation and 5b1b3

relevance functions, 5b1b4

Innovations in	5b2
publishing,	5b2a
retrieval,	5b2b
and user services	5b2c
are eligible,	5b3
Projects may be	5b4
design and development efforts oriented to	5b4a
specific innovation	5b4a1
in the	5b4a1a
technology,	5b4a1a1
organization and	5b4a1a2
management	5b4a1a3
of scientific communications,	5b4a1b
or operational experiments designed to	5b4b
test and validate	5b4b1
hypotheses	5b4b1a
or promising innovations	5b4b1b
developed from prior research efforts,	5b4b2
Areas of special interest include:	5b5
improved retrieval techniques and capabilities,	5b5a
innovative products and services	5b5b
which can be offered by	5b5b1
libraries and	5b5b1a
information service centers,	5b5b1b
cost-effective	5b5c

information storage media and 5b5c1

retrieval/delivery technology, 5b5c2

paperless publication by means of micrographic and electronic media, 5b5d

cost-effective resource sharing, 5b5e

cooperative endeavors in document delivery services, and 5b5f

user-oriented publications, 5b5g

Innovations and improvements may be 5b6

in the 5b6a

technology, 5b6a1

organization, or able 5b6a2

management 5b6a3

of 5b6b

scientific communication and 5b6b1

technology transfer 5b6b2

services, 5b6c

Category 3: Tests and analyses of marketing strategies and techniques 5c

Innovative marketing strategies for disseminating R&D results need to be investigated in order to 5c1

create user awareness, and to 5c1a

broaden 5c1b

the 5c1b1

user and 5c1b1a

economic 5c1b1b

bases of 5c1b2

publishing, 5c1b2a

retrieval, and 5c1b2b

library 5c1b2c

operations, 5c1b3

Marketing strategies should be based upon the concept 5c2

that "information" is a "commodity," and 5c2a

that innovative marketing theory and practices could expand sales or enhance dissemination, 5c2b

OSIS intends to support 5c3

studies of 5c3a

the effects of marketing variables such as 5c3a1

price, 5c3a1a

promotion, 5c3a1b

product design, and 5c3a1c

distribution methods 5c3a1d

on 5c3a2

use and value of 5c3a2a

scientific and technical information products and services, 5c3a2b

The program also intends to support 5c3b

marketing research on 5c3b1

utilization of 5c3b1a

scientific and technical information and 5c3b1a1

the identification of 5c3b1b

new applications, 5c3b1b1

new users, and 5c3b1b2

required new services; 5c3b1b3

building and testing of models for simulating the 5c3b2

STI marketplace or 5c3b2a

some significant proportion of the overall market; and 5c3b2b

forecasting studies related to future markets for STI, 5c3b3

Studies of 5c3c

purchasing and consumer behavior relevant to 5c3c1

designing marketing strategies and techniques 5c3c1a

for scientific communications 5c3c1b

may also be supported, 5c3d

Studies may be 5c4

conceptual or empirical and 5c4a

may be directed toward 5c4b

any segment of the 5c4b1

information marketplace or 5c4b1a

service operations 5c4b1b

such as 5c4b2

journals, 5c4b2a

abstracting and indexing services, or 5c4b2b

libraries, 5c4b2c

Category 4: Economic characteristics of scientific and technical information communication 5d

One of the major barriers 5d1

to the development of 5d1a

cost effective and user responsive 5d1a1
scientific and technical communication services 5d1a2
has been the very limited amount of data 5d1b
on the economics of information transfer. 5d1b1
The program intends to support projects desiged 5d2
to generate concepts and data 5d2a
relevant to policy and management decisions or 5d2a1
to develop and test 5d2b
detailed input/output matrices and simulation models 5d2b1
of the STI marketplace. 5d2b2
Efforts should contribute to 5d3
the development of 5d3a
a comprehensive data base 5d3a1
for use in 5d3b
describing and analyzing the 5d3b1
transactions, 5d3b1a
economic character, and 5d3b1b
markets 5d3b1c
for STI products and services. 5d3b2
Results should help 5d4
to identify 5d4a
new product opportunities and 5d4a1
new market potentials 5d4a2
in scientific and technical communication. 5d4b
Suitable topics of investigation include 5d5

supply/demand relationships, 5d5a

direct and indirect (external) costs and benefits of services, 5d5b

cost functions and economy of scale, 5d5c

consumer preference or indifference curves, and 5d5d

price elasticity. 5d5e

Category 5: Performance evalution of services and systems 5e

Data, literature and services continue to expand rapidly. 5e1

Producers, purchasers, and users need 5e2

methods and data for 5e2a

evaluating what they are 5e2b

offering, 5e2b1

buying 5e2b2

or using. 5e2b3

OSIS is prepared to support studies 5e3

designed to 5e3a

measure the performance of 5e3b

scientific and technical communication enterprises 5e3b1

in the United States . 5e3b2

Studies may include 5e4

analyses of STI services 5e4a

in relation to their 5e4b

economic viability, 5e4b1

cost/benefit ratios, 5e4b2

growth rates, 5e4b3

user acceptability, 5e4b4
potential market expansion, 5e4b5
return on investment, 5e4b6
impacts on users' share of the market, 5e4b7
impacts on scientific and technical activities, and 5e4b8
other related variables. 5e4b9
Criteria used 5e5
for evaluating 5e5a
specific information systems or services 5e5a1
may include 5e5b
cost (of 5e5b1
design, 5e5b1a
development 5e5b1b
and operations), 5e5b1c
content of data bases 5e5b2
(scope, 5e5b2a
quality, 5e5b2b
currency), 5e5b2c
retrieval performance 5e5b3
(speed, 5e5b3a
format, 5e5b3b
usability, 5e5b3c
efficiency and ease of operation), 5e5b3d
expandability and linkage to other systems, 5e5b4
reliability, 5e5b5

accessibility, and 5e5b6

value to users. 5e5b7

The purpose of such investigations is 5e6

the development of generalized performance evaluation systems 5e6a

for use by 5e6b

information services managers, 5e6b1

policy makers, 5e6b2

and users. 5e6b3

Category 6: Energy R&D information 5f

The energy R&D literature 5f1

is abstracted and indexed 5f1a

by many services, 5f1a1

all of which must be consulted by an investigator seeking comprehensive coverage. 5f1a2

This search for information is costly and time-consuming, 5f1b

sometimes to the point that investigators find the effort is not worthwhile. 5f1b1

OSIS intends to support projects 5f2

to develop and test 5f2a

ways of making 5f2b

the Nation's existing energy-related R&D information and data resources 5f2b1

quickly available and easily retrievable to users. 5f2b2

Projects must be 5f3

empirical, 5f3a

pragmatic,	5f3b
of short duration,	5f3c
and addressed	5f3d
to the	5f3d1
technical,	5f3d1a
organizational and	5f3d1b
managerial functions	5f3d1c
of the energy R&D information transfer	5f3d2
from originator to ultimate user.	5f3d2a
Projects will be considered for:	5f4
* development and comparative evaluation of	5f4a
methodologies for	5f4a1
indexing the energy literature and for	5f4a1a
retrieving energy information	5f4a1b
from existing mechanized information systems.	5f4a2
* feasibility and preliminary design studies of	5f4b
a national energy R&D information center and	5f4b1
specialized information and data centers	5f4b2
for specific energy fields.	5f4b2a
* preliminary design studies of	5f4c
the connecting and interfacing systems needed	5f4c1
for a	5f4c2
national and international	5f4c2a
energy R&D	5f4c2b
information and data	5f4c2c

network, 5f4c3

The techniques 5f5

developed and tested in the proposed projects 5f5a

must also be applicable in other national need areas 5f5b

such as the 5f5b1

environment, 5f5b1a

natural resources, 5f5b1b

population, and 5f5b1c

food 5f5b1d

information fields, 5f5b2

Category 7: Data tagging -- design studies and operational experiments 5g

Direct access 5g1

to the quantitative data of science and technology 5g1a

cannot be assured until 5g1b

improved methods are developed for 5g1c

identifying the data content of papers published 5g1d

in primary journals and in 5g1d1

technical reports, 5g1d2

A natural first step toward the resolution of this problem 5g2

is to provide a concise description 5g2a

of the data content of a paper 5g2a1

through the incorporation 5g2b

of well-understood data descriptors or tabs 5g2b1

into the article and/or its abstract, 5g2c

OSIS is prepared to consider proposals for 5g3

intensive design studies 5g3a

aimed at the detailed specification of such innovations, 5g3a1

Proposals for operational experiments 5g3b

can be considered for those organizations or groups 5g3b1

that are already clearly beyond the design stage of data identification, 5g3b1a

Projects involving energy-related data tagging 5g4

will be given priority consideration, 5g4a

Category 8: Editorial processing center -- operational experiments 5h

An editorial processing center (EPC) is a mechanism 5h1

for combining small publishing operations 5h1a

to achieve a scale great enough 5h1a1

for cost-effective investment in modern technology. 5h1a2

A fully implemented EPC is conceived as 5h1b

a computer-based complex 5h1b1

through which are channeled all communications between 5h1b2

an editor 5h1b2a

and the 5h1b2b

authors, 5h1b2b1

referees, 5h1b2b2

and printers 5h1b2b3

with whom he interacts. 5h1b2c

A study report has recently been completed 5h2

which described in detail 5h2a

four possible EPC* configurations 5h2a1

and discusses the technical and economic feasibility of each, 5h2b

OSIS is prepared to consider proposals for 5h3

the testing of one or more operational EPCs. 5h3a

A proposed EPC 5h4

may be an independent entity, 5h4a

either proprietary or not-for-profit, 5h4a1

or it may be established within the frame-work of an existing institution, 5h4b

The proposed 5h5

configuration need not conform to one of those described in the report; 5h5a

however, the proposal should discuss in similar depth 5h5b

the services to be provided, 5h5b1

the community of users, 5h5b2

and provide details related to coordinated financing. 5h5b3

Category 9: Assessment of the health of the scientific and technical communication system of the United States 5i

OSIS intends to provide support for an annual assessment of the health of the scientific and technical communication system of the United States. 5i1

The assessment should include 5i2

development of national performance measures 5i2a

for evaluating effectiveness 5i2a1

in the 5i2a1a

acquisition,	5i2a1a1
publication,	5i2a1a2
distribution,	5i2a1a3
retrieval,	5i2a1a4
and use	5i2a1a5
of scientific and technical information;	5i2a1b
application of these measures	5i2b
to indicate changes in the social benefits	5i2b1
rendered by communication services;	5i2b1a
identification of recent major advances and losses	5i2c
in communication capabilities; and	5i2c1
identification and analyses of serious, persistent problems	5i2d
related to dissemination of scientific and technical information.	5i2d1
Proposals should include two phases:	5i3
(1) a design phase in which performance measures and research methods will be developed and pilot-tested; and	5i3a
(2) the field research phase in which data will be collected, analyzed, and prepared for publication.	5i3b
Field work should be limited to assessment in mid-calendar year 1975.	5i4
Calendar year 1975 will serve as the bench mark for subsequent annual assessments.	5i4a
Therefore, a wide range of variables should be included.	5i4b
Emphasis should be on the health of operational services, including those by public, private non-profit, and profit-making organizations.	5i5

OSIS may support more than one project at the design level, but will not fund more than one subsequent field effort. 5i6

Offerors should know that the National Science Foundation is supporting a related study on the development of indicators of the supply and demand for scientific and technical information. Initial results of the indicators study will be available by summer 1975 to any organization conducting the assessment study. Indicator data are seen as input data for helping with the assessment task. 5i6a

OSIS staff will take an active role in working with the organization during the assessment field work. 5i6b

PROPOSAL FORMAT 5j

Cover Sheet 5j1

Abstract 5j2

Not to exceed 250 words 5j2a

Summary of the project, including 5j2b

The objctive, 5j2b1

approach or methodology, and 5j2b2

anticipated impact. 5j2b3

Narrative 5j3

Concise and to the point, 5j3a

leaving less essential details for appndices. 5j3a1

Should include sections on: 5j3b

Introduction 5j3b1

Short statement of project goals and detailed objectives; 5j3b1a

anticipated benefits and impact of the proposed activity on STI services and/or use; 5j3b1b

relation to the present state of knowledge and activity in the fild, 5j3b1c

previous work, and 5j3b1c1

work-in:progress elsewhere; an 5j3b1c2

a bibliography of pertinent literature. 5j3b1d

Study Plan 5j3b2

A majorporton f the narrative should be devoted to the propsed research design and methodology. 5j3b2a

The general plan of ork ad of specific procedures to be folowed should be clearly specified. 5j3b2b

Organization and Management Plan. 5j3b3

Schedules and detailed work statements should be prepared, containing staff assignments and showing the necssary steps for achieving the project objectives. 5j3b3a

Estimates of time to reach each step should be included. 5j3b3a1

It is essential that there be identified a single person with responsibiliy for planning, coordinating, supervising, and integrating the work. 5j3b3b

When organiztional collaborative arrangements are utilized, the motivation and rationale for such collaboration should be dicussed, along with details regarding how the work will be coordinated. 5j3b3c

The investigators are expcted to submit a phased program plan with checkpoinnts and milestones ofprogress, including two or three reviews bythe Fundtion's staff. Relationships between tasks, personnel, and budget items must be shown in a single exhibit. 5j3b3d

Dissemination of Results 5j3b4

Institutional Resources and Related Programs 5j3b5

Personnel	5j3b6
Budget	5j4
Appendices (as appropriate)	5j5

DCE 27-FEB-75 18:42 25509

Residue from preparing NSF-OSIS SSSC proposal (24758,)

(J25509) 27-FEB-75 18:42;;;; Title: Author(s): Douglas C. Engelbart/DCE; Sub-Collections: SRI-ARC; Clerk: DCE;

JBP 27-FEB-75 19:19 25510

RFC Index for 1-JUN-74 to 1-FEB-75

This note is an index of ARPA Network Working Group Requests for Comments issued in the period 1-JUN-74 through 1-FEB-75.

The distribution of RFCs is the responsibility of the Author.

Distribution lists for RFCs are available online:

[OFFICE-1]<NETINFO>LIASION.TXT

is a list of U.S. Mail address of the technical liaisons.

[OFFICE-1]<NETINFO>LIASION-SNDMSG.TXT

is a list of ARPANET sndmsg address of the technical liaisons.

[SRI-ARC]<POSTEL>RFC.DISTRIBUTION-LIST

is a list of ARPANET sndmsg address of persons who have expressed interest in RFCs and includes the technical liaisons.

Currently it is possible to store online copies of RFCs as text files at Office-1, authors who wish to do this should contact Jon Postel.

RFC numbers are assigned by Jon Postel.

SNDMSG address: POSTEL at SRI-ARC
Journal ident: JBP
Phone number: (415) 326:6200 x3718

Some RFCs are online as Journal items and many of the recent RFCs are online as sequential ASCII text files at Office-1 with file names of the form <NETINFO>RFCnnn.TXT, where nnn is the RFC number.

Files may be pulled from Office-1 using the Server FTP login parameters USER= ANONYMOUS, PASS= your initials, ACCT= space.

RFC 640, Journal 30843
Date: 5-JUN-74
Author: Nancy Neigus, Jon Postel, Ken Pogran
Title: Revised FTP Reply Codes

RFC Index for 1-JUN-74 to 1-FEB-75

RFC 641, Journal
Date:
Author:
Title: NOT ISSUED

RFC 642, Journal 30872
Date: 5-JUL-74
Author: Jerry Burchfiel
Title: Ready Line Philosophy and Implementation

RFC 643, Journal 30873
Date: 5-JUL-74
Author: Eric Mader
Title: Network Debugging Protocol

RFC 644, Journal 30874
Date: 22-JUL-74
Author: Robert Thomas
Title: On the Problem of Signature Authentication for Network Mail

RFC 645, Journal 30899
Date: 26-JUN-74
Author: David Crocker
Title: Network Standard Data Specification Syntax

RFC 646, Journal
Date:
Author:
Title: NOT ISSUED

RFC 647, Journal 31117
Date: 12-NOV-74
Author: Michael Padlipsky
Title: A Proposed Protocol for Connecting Host Computers to ARPA-Like Networks via Directly Connected Front End Processors

RFC 648, Journal
Date:
Author:
Title: NOT ISSUED

RFC 649, Journal
Date:
Author:
Title: NOT ISSUED

RFC Index for 1-JUN-74 to 1-FEB-75

RFC 650, Journal
Date:
Author:
Title: NOT ISSUED

RFC 651, Journal 31154
Date: 25-OCT-74
Author: David Crocker
Title: Revised Telnet Status Option

RFC 652, Journal 31155
Date: 25-OCT-74
Author: David Crocker
Title: Telnet Output Carriage Return Disposition Option

RFC 653, Journal 31156
Date: 25-OCT-74
Author: David Crocker
Title: Telnet Output Horizontal Tab Stops Option

RFC 654, Journal 31157
Date: 25-OCT-74
Author: David Crocker
Title: Telnet Output Horizontal Tab Disposition Option

RFC 655, Journal 31158
Date: 25-OCT-74
Author: David Crocker
Title: Telnet Output Form Feed Disposition Option

RFC 656, Journal 31159
Date: 25-OCT-74
Author: David Crocker
Title: Telnet Output Vertical Tab Stops Option

RFC 657, Journal 31160
Date: 25-OCT-74
Author: David Crocker
Title: Telnet Output Vertical Tab Disposition Option

RFC 658, Journal 31161
Date: 25-OCT-74
Author: David Crocker
Title: Telnet Output Line Feed Disposition Option

RFC 659, Journal 31177
Date: 18-OCT-74

Author: Jon Postel
Title: Announcing Additional Telnet Options

RFC 660, Journal 31202
Date: 23-OCT-74
Author: Dave Walden
Title: Some Changes to the IMP and the IMP/Host Interface

RFC 661, Journal 31203
Date: 23-NOV-74
Author: Jon Postel
Title: Protocol Information

RFC 662, Journal 31386
Date: 26-NOV-74
Author: Raj Kanodia
Title: Performance Improvement in ARPANET File Transfers from Multics

RFC 663, Journal 31387
Date: 29-NOV-74
Author: Raj Kanodia
Title: A Lost Message Detection and Recovery Protocol

RFC 664, Journal
Date:
Author:
Title: NOT ISSUED

RFC 665, Journal
Date:
Author:
Title: NOT ISSUED

RFC 666, Journal 31396
Date: 26-NOV-74
Author: Michael Padlipsky
Title: Specification of the Unified User-Level Protocol

RFC 667, Journal 31422
Date: DEC-74
Author: Robert Thomas
Title: BBN Host Ports

RFC 668, Journal
Date:
Author:

Title: NOT ISSUED

RFC 669, Journal 31435
Date: 4-DEC-74
Author: Doug Dodds
Title: November 1974, Survey of New-Protocol Telnet Servers

RFC 670, Journal
Date:
Author:
Title: NOT ISSUED

RFC 671, Journal 31439
Date: 6-DEC-74
Author: Richard Schantz
Title: A Note on Reconnection Protocol

RFC 672, Journal 31440
Date: 6-DEC-74
Author: Richard Schantz
Title: A Multi-Site Data Collection Facility

RFC 673, Journal
Date:
Author:
Title: NOT ISSUED

RFC 674, Journal 31484
Date: 12-DEC-74
Author: Jon Postel, Jim White
Title: Procedure Call Documents -- Version 2

RFC 675, Journal 31505
Date: 16-DEC-74
Author: Vint Cerf, Yogen Dalal, Carl Sunshine
Title: Specification of Internet Transmission Control Program

RFC 673, Journal
Date:
Author:
Title: NOT ISSUED

RFC 677, Journal 31507
Date: 27-JAN-75
Author: Paul Johnson, Robert Thomas
Title: The Maintenance of Duplicate Databases

RFC Index for 1-JUN-74 to 1-FEB-75

RFC 678, Journal 31524
Date: 19-DEC-74
Author: Jon Postel
Title: Standard File Formats

RFC 700, Journal 31020
Date:
Author: Eric Mader, William Plummer, Raymond Tomlinson
Title: A Protocol Experiment

RFC 701, Journal
Date: AUG-74
Author: Doug Dodds
Title: August, 1974, Survey of New-Protocol Telnet Servers

RFC 702, Journal
Date: 25-SEP-74
Author: Doug Dodds
Title: September, 1974, Survey of New-Protocol Telnet Servers

RFC Index for 1-JUN-74 to 1-FEB-75

(J25510) 27-FEB-75 19:19;;;; Title: Author(s): Jonathan B. Postel/JBP; Distribution: /SRI-ARC([INFO-ONLY]) NSW([INFO-ONLY]) NLG([INFO-ONLY]) NSAG([INFO-ONLY]) ; Sub-Collections: SRI-ARC NSW NLG NSAG; Clerk: JBP; Origin: < POSTEL, RFC-INDEX.NLS;2, >, 27-FEB-75 19:11 JBP ;;;;.HRJM=72;####;

CHI 27-FEB-75 20:09 25511

Some notes on user sequence generators

Sent via SNDMSG to Tugender and Mandell at ISI. Thought you might be able to gleen something for documentation from this. I included <nls, seqgen,>.

Some notes on user sequence generators

Dick and Ron, it seems that the documentation on making sequence generators was dropped from our supported documentation long ago (no one used it) and I can't find a copy of it anywhere. I would be able to find it on an old archive tape if I had a few weeks, but I don't so I will write you a few notes and send you some related documentation instead. I recommend that you read the documentation on content analysis that is enclosed and then read the source code that is enclosed which consists of the system's sequence generator (SSEQGEN) and an example of a user sequence generator (PJSEQG) for printing out Journal mail by following citations to other files. In reading this code, you should remember that: 1

The sequence generator has a co-routine linkage to its caller. This is effected by the routines SPORT and SEQGEN. (This code was written before co-routines were added to L10.) They do a stack switch to effect the co-routine transfer. OPENSEQ sets up the new stack and port machinery and CLOSESEQ reclaims it. OPENSEQ takes as an argument the address of a sequence generator to use or 0, meaning the default system sequence generator. The address of a sequence generator work area is used as the port handle that connects the co-routines. Thus, NLS calls SEQGEN to get the next in the sequence. SEQGEN does a co-routine call on the real sequence generator which computes the next value and calls SPORT to do a port call back to its caller. 1a

The sequence generator is called in three modes 1b

1) initialization (called from openseq), 1b1

2) get next in sequence, and 1b2

3) termination (called from closeseq). 1b3

The sequence generator may be asked (via a parameter to OPENSEQ and viewspecs) to invoke a user content filter as the final test as to whether a statement should appear in the sequence. 1c

The procedure SEND calls SPORT after fidling the work area so that the string passed to it will appear to be the next statement in the sequence. 1d

A user sequence generator will often find it useful to call the system sequence generator SSEQGEN or to call OPENSEQ, SEQGEN, and CLOSESEQ for subordinate sequences. 1e

The size of the system sequence generator is 724 (octal) [openseq to relsgw]. The addition of other utility functions and the print journal mail sequence generator PJSEQG (533 octal words) brings the total for the enclosed file to 2463 (octal) words. 2

CHI 27-FEB-75 20:09 25511

Some notes on user sequence generators

(J25511) 27-FEB-75 20:09;;;; Title: Author(s): Charles H. Irby/CHI;
Distribution: /POOH([INFO-ONLY]) NDM([INFO-ONLY]) ;
Sub-Collections: SRI-ARC; Clerk: CHI; Origin: < IRBY,
SEQ.NLS;2, >, 21-FEB-75 17:27 CHI ;;;;####;

CHI 27-FEB-75 21:58 25512

Phone conversation between Irby and Millstein

Sent via SNDMSG to Millstein, Warshall, Sattley, Balzer, Crocker,

CHI 27-FEB-75 21:58 25512

Phone conversation between Irby and Millstein

This memo documents phone conversations between Charles Irby and Bob Millstein on 13-Feb-75 and 14-Feb-75. A number of details were discussed which were not completely clarified at the last review meeting. My understanding of the resolution of these details follows: 1

File name recognition 1a

The fields forming the file name will be an ordered list. Thus a.b is not the same as b.a. Note that this is a deviation from MCA's original desire to have the project leader specify whether or not the fields form an ordered list. It appears to us at ARC that having the fields ordered is necessary to properly support file name completion. 1a1

Bob described a possible form of field completion which would not take into account the user's current working directory (scope -- the prefix that is automatically added to his file name to form complete file names) or the portion of the file name he has already specified. It would, rather, search a list of all fields used by this user (his project? all of NSW?) and would do completion based on uniqueness within this list. 1a2

We suspect that the file system being built must be inverted on field names and must be organized for hash code lookups only. This is why doing field completion is so difficult. Only time will tell how useful field completion is but we suspect that the majority of users will use scope control to keep the names they must specify simple and that often they will be one field long, from the user's viewpoint. For these files, there will be no file name recognition because there is no real field recognition. 1a2a

tool startup 1b

The FE will open the packages specified in the grammar but the WM will create the processes and introduce the tool processes to the FE. The WM will get the process names from its interactive tool descriptor or from the grammar. 1b1

The list of processes and packages are contained in the grammar and form a linked list which is pointed to by the grammar dispatch record at the beginning of the grammar. Along with each process name (a string of characters which has an L10 string header and ends with a 0 character) will also be stored a ten-bit hash code for it. 1b1a

reporting slueing to WM 1c

Phone conversation between Irby and Millstein

The FE will allow users to have more than one active tool. At any point in time he will be interacting with only one of these. This tool is called the current tool. The FE will not report to the WM when the user makes another of his active tools the current tool. If the FE crashes while the user has more than one active tool, then on restart it will find out from the WM which tools were active for each user but not which was the current tool. In this case, the user will be placed in the NSW-EXEC grammar from which he may resume whichever tool he wishes. 1c1

Files created during tool interactions 1d

Bob did not want to support temporary files as outlined in my tool interaction memo but rather suggested that this could be done entirely by the tools using only local files. He felt the files could be moved and converted by t4e interacting tools. I dont understand how this would work but am willing to think about it. Some immediate questions come to mind: 1d1

How did the tools get process handles for the file package process? 1d1a

How do the tools know what type of conversions to perform? The structure type is not known for a local file -- that is an NSW file system phenomenon. 1d1b

Even if the tools could deduce the structure type, how do they know what conversion routines exist? Isn't that something the WM knows? 1d1c

Help returns from a tool when it is not the current tool 1e

If the user has several active tools and an outstanding call to a tool that is not now the current tool HELPs, the the FE will notify the use9 that the tool has something to say to him but will not present any output from the tool until the user makes it the current tool. This should avoid any ambiguous situations. I need to find out more about what this means in PCP terms, but will assume for the time being that there will not be severe problems here. 1e1

tool list updating in the FE 1f

The FE will want a list of available tools for the user so it can tell him when he asks ans so it can allow recognition on tool names. If the project leader changes the tools available to the user, the WM can call a primitive in the FE and update its tool list for this user. The WM will perform additional

tests to see if the tool use is valid for this user. The list in the FE is merely for user interaction and in no way eliminates the need for the WM to perform its access control function. 1f1

The FE will also make available a primitive to update the user's interaction profile. This is needed by the user-profile tool. 1f2

root process for a tool 1g

After some discussion, Bob and I agreed that for July-75 there will be a single PCP process associated with a grammar (a tool), but that this was not a good long-term design. There will exist in the future grammars for which more than one top level process will be needed, although these will not be common. The CML already allows for this and the WM will in the future. 1g1

getgrammar, getprofile 1h

The WM will provide a getGrammar and a getprofile primitive for use by the FE, the Help process, and the User-profile tool. 1h1

User programs 1i

With respect to NLS user programs (see my user programs memo), Bob does not think this should involve the WM. He feels that the project leader can place restrictions on their use by limiting access to the files containing the programs and their grammars. He feels the FE and the tool should accomplish this without involving the WM. We do not understand exactly how to pull this off but are willing to think about it. 1i1

WM going to user for help 1j

If the tool makes a call on the WM with an ambiguous filename, the WM should do a HELP return to the tool rather than going straight to the user at his FE, at least for July-75. It seems to me imperative that things work this way since the file name could easily have been generated by the tool in which case the user will know nothing about it. Bob suggested adding a boolean flag to appropriate calls which would tell the WM to HELP or call a procedure in the FE. It was left undecided exactly how this would be handled. 1j1

TBH local files and crashes 1k

The need to preserve WM or tool knowledge of local files across

Phone conversation between Irby and Millstein

TBH crashes was discussed. We agreed that it was necessary but still a bit fuzzy. 1k1

NLS needs a way to allow the user to give it a Create File command then add some text to the file and have the TBH crash and recover and still have his file when he next asks for it. As we understand the current WM primitives this need is not met, but clearly should be by july-75. 1k1a

Semaphor needed for NLS 11

NLS would like to provide its users with an inter-lock capability such that if a user (with replace access to a file) starts editing the file (which he has a local copy of) NLS will set a semaphor on the original file such that other users who attempt to modify their copies of the same file can be told that user X is already modifying the file. Please note that the changes a user wishes to make can easily cross login session boundaries. Thus the semaphore should be set when he start his editing and reset when he either decides his changes are consistent and complete and replaces the original file with his update or throws his edits away. 111

Thus, we need a set/reset/test semaphore capability for each file. This may not be the only semaphor needed per file by the file system and tools. Bob mentioned that they already had planned on a semaphor for transaction files but was reluctant to use that semaphor for cross-session interlocks. He also specified that they were thinking of setting such a semaphor whenever a user with replace access took a copy of a file from the WM. This may be appropriate for some tools but will not work very well with NLS. We would be willing to use such a scheme and simply reset the semaphor immediately if we don't want it set, but would want to leave it set until the user released it not just when he logged out. 112

Suspending and resuming a tool 1m

The user will be able to suspend and resume execution of outstanding calls on tool processes. This will be done entirely by the FE and will not effect the WM. 1m1

CHI 27-FEB-75 21:58 25512

Phone conversation between Irby and Millstein

(J25512) 27-FEB-75 21:58;;;; Title: Author(s): Charles H. Irby/CHI;
Distribution: /NPG([INFO-ONLY]) RWW([INFO-ONLY]) WEC([
INFO-ONLY]) DLS([INFO-ONLY]) LAC([INFO-ONLY]) ;
Sub-Collections: SRI-ARC NPG; Clerk: CHI; Origin: < IRBY,
MILL-CONV,NLS;1, >, 18-FEB-75 17:34 CHI ;;;;####;

YL 27-FEB-75 23:56 25513

Friendly note at the milestone.

Farewell sri-arc. 1

Dear Doug: 1a

Even here in far away Norway (close to the North Pole), there are people who have sad feelings about SRI-ARC going away. 1b

In the very informal way it all has happened, I have personally learnt a lot from the use I have been able to make of it (limited chiefly by myself) 1c

I am grateful for the possibilities which have generously been made available . Let me take this opportunity and say to yourself and all my other friends at SRI, that my heart beats a little extra for you over there today. 1d

Without too much inside information, and therefore uncertain of the details, I view the event of removing the machine as the turn of a successful era, now a foundation for further development. Let me mention that if you are looking for friends interested in harnessing and contributing to technological progress for improved communications and other needs of people, you have one right here in Norway, - just one message away. 1e

I hope , therefore to continue to exchange viewpoints, ideas and inspiration, and possibly to be able to enter into active collaborative efforts in the time to come, as this world rids itself of the obstacles of geographical distance. 1f

Sincerely Yngvar (= Lundh , Kjeller, Norway) 1g

YL 27-FEB-75 23:56 25513

Friendly note at the milestone.

(J25513) 27-FEB-75 23:56; Title: Author(s): Yngvar Lundh/YL;
Distribution: /DCE; Sub-Collections: NIC; Clerk: YL;

JDH 3-MAR-75 22:56 25515

FANFARE, please

Hellll00000000 there

JDH 3-MAR-75 22:56 25515

FANFARE, please

(J25515) 3-MAR-75 22:56;;;; Title: Author(s): J. D. Hopper/JDH;
Distribution: /HGL([ACTION]) CHI([INFO-ONLY]) ; Sub-Collections:
SRI-ARC; Clerk: JDH;

JBP 4-MAR-75 15:09 25516

PCP Questions from Braden

4-MAR-75 12:20:45-EDT,5278;00
Mail from USC-ISI rcvd at 4-MAR-75 1220-EDT
Date: 4 MAR 1975 0919-PDT
From: CCN at USC-ISI
Subject: QUESTIONS ON PCP
To: WHITE at BBNB
cc: BRADEN at CCN, POSTEL at BBNB 1

Jim, As you are probably aware, CCN will be a TBH. Therefore, I have been trying to decipher the White/Postel Rosetta Stone. I readily admit to a large confusion, and would like to ask some questions to help get my head straight. Here goes: 1a

1. It is difficult to keep straight all the different kinds of handles. Collecting the definitions together in one place in the document might help. 1b

a. PCPHST refers to "connection handles", although it does not define the term. Is that the receive socket number (special case of PORT#) referred to in 2b1? 1b1

b. The purpose and definition of the PCH of PMP needs elaboration in the document. Also, the procedure (primitive?) PCHAN used ,3b1d1a of PMP is not defined anywhere that I could find. 1b2

2. I am confused about the nature of the Procedure Interface Package. It is a set of procedures for calling procedures, apparently, but that is recursive; you need PIP to invoke PIP. Do you really intend PIP to contain PCP-callable procedures? there clearly must be some low-level CALPRO interface (like SyCALL in Tenex). Is that the PIP, or is in addition to the PIP? There doesn't seem to be much point to having the PIP as real callable procedures in that case.... 1c

3. When a new process is created, apparently a "primary" channel is opened automatically. The processes can open secondary channels using CRTPHYCHNEND , etc. Right? Could one call the primary channel a "master" or "control" channel? 1d

a. DELPRC is defined to invalidate the poh for the primary channel. Wouldn't it invalidate all channels open to that process? 1d1

b. Because the primary channel is opened intrinsically during the creation of a new process, there is no mechanism to determine its type. Is there a standard assumption defined somewhere? 1d2

PCP Questions from Braden

4. The PCP document says that as a result of a CRTPRC the new process receives the POH of his creator. It is not clear to me how this happens. If PMP does it, the new process is inferior, and can refer to his superior by -1. ... 1e

OOps, that is not right. He refers to his superior by PH=-1. How did POH get into this? I must be confused between levels of abstraction again. I guess I know what the answer must be, but I7d like to hear your explnaation. 1f

5. I don't really understand the meaning or function of the "generic name PRCNAME". 1g

6. The definition of PRCLOC* seems either to be very Tenex-dependent, or else to include some semantics that are not stated explicitly. The concept of "job" or "job number" is never defined anywhere. Perhaps PRCLOC* should contain a host-dependent string after <host>. 1h

JBP 4-MAR-75 15:09 25516

PCP Questions from Braden

(J25516) 4-MAR-75 15:09;;; Title: Author(s): Jonathan B. Postel/JBP; Distribution: /JEW([INFO-ONLY]) ; Sub-Collections: SRI-ARC; Clerk: JBP;

CHI 4-MAR-75 19:01 25517

Smoking

The recent passage by Palo Alto of a "no smoking" ordinance prompts me to make a recommendation to ARC that I have felt was overdue for a long time. The recomendation is that ARC personnel refrain from smoking during meeting, especially in the parsely room. In addition, I recommend that a portion of the open work area be set aside for smokers and that sufficiently strong exhaust fans be installed to keep the smoke from spreading to the rest of the area. 1

CHI 4-MAR-75 19:01 25517

Smoking

(J25517) 4-MAR-75 19:01;;;; Title: Author(s): Charles H. Irby/CHI;
Distribution: /RWW([ACTION]) DCE([ACTION]) JCN([ACTION])
SRI-ARC([INFO-ONLY]) ; Sub-Collections: SRI-ARC; Clerk: CHI;

Sick Ident Program

I would like to stress AGAIN that we are in trouble with respect to the identfile program. NLS 7 has now gone away and Marcia is having some offthe same problems I was having. At this point there is no guaranteed way to maintain the running identfile successfully. This needs someone's attention very soon. Since the priorities are overwhelming at the moment I do not know what the answer is, but emphasize again that there is an immediate problem. 1

Sick Ident Program

(J25518) 5-MAR-75 01:38;;;; Title: Author(s): Elizabeth J. (Jake) Feinler/JAKE; Distribution: /DCE([ACTION]) JCN([ACTION]) RWW([ACTION]) KJM([INFO-ONLY]) MLK([INFO-ONLY]) JDH([INFO-ONLY]) FEEDBACK([INFO-ONLY]) ; Sub-Collections: SRI-ARC FEEDBACK; Clerk: JAKE;

DVN 5-MAR-75 12:38 25519

Problems with Control Characters, Processing Commands Branches at BBN

Running through a TIP only to BBN-B you do not seem to be able to enter control characters into a file via <CTRL-V><CTRL- >;you do not seem able to redefine a user's control characters via the Useroptions Subsystem and where user options have been set to special control characters in the past, they have been reset to the defaults;last and worst, any attempt to process a commands branch puts you in exec with the message "ILLEGAL INSTRUCTION JSYS SQSVWS+107 = 104000,,521" etc. etc.

1

DVN 5-MAR-75 12:38 25519

Problems with Control Characters, Processing Commands Branches at BBN

(J25519) 5-MAR-75 12:38;;;; Title: Author(s): Dirk H. Van Nouhuys/DVN; Distribution: /FEED([ACTION]) ; Sub-Collections: SRI-ARC; Clerk: DVN;

CHI 5-MAR-75 20:45 25520

Insert date command

In reply to your message of 2-MAR-75 20:21 Susan/FEED
Journal: (31941,)
Subject: acknowledgement of (25449,) Insert date command

1

Date need not be a second level commandword in the Insert Date command. JDH should change this for all NLS's in the world. CHI 1a

CHI 5-MAR-75 20:45 25520

Insert date command

(J25520) 5-MAR-75 20:45;;; Title: Author(s): Charles H. Irby/CHI;
Distribution: /FEED([INFO-ONLY]) JDH([INFO-ONLY]) ;
Sub-Collections: SRI-ARC; Clerk: CHI;

KEV 5-MAR-75 20:48 25521

4700 questions

larry, 1

i was speaking to tom hamrick of sai earlier today and he did not think that the 4700 supported a process tree structure. if this is indeed the case, the implementation of pcp and interactive debuggers along the lines that we talked about earlier becomes more difficult, if not impossible. any comments? 1a

he also mentioned something called afold. is this the thing you are going to be sending me? 1b

lastly, can i get a copy of the b3500 concepts manual? 1c

KEV 5-MAR-75 20:48 25521

4700 questions

(J25521) 5-MAR-75 20:48;;; Title: Author(s): Kenneth E. (Ken) Victor/KEV; Distribution: /LAC([ACTION]) ; Sub-Collections: SRI-ARC; Clerk: KEV;

KIRK 5-MAR-75 23:43 25522

KKELLEY at BBN not KELLEY

My new directory is KKELLEY. If you have sent any messages to KELLEY, they have not come to me. Ident is still KIRK. 1

KIRK 5-MAR-75 23:43 25522

KKELLEY at BBN not KELLEY

(J25522) 5-MAR-75 23:43;;;; Title: Author(s): Kirk E. Kelley/KIRK;
Distribution: /SRI-ARC([INFO-ONLY]) ; Sub-Collections: SRI-ARC;
Clerk: KIRK;

CHI 5-MAR-75 23:59 25523

scrolling in nls

Jeanne Beck asked in 31898 why scrolling would be expensive in DNLS. If implemented simply (That is, such that " would only work reasonably to undo the last LINEFEED), it would not be at all expensive and would be very easy to implement. The scroll-forward (LINEFEED) part is simple nad not expensive. The scroll-backward could be very expensive if the user was allowed to do it at any point such that the image now in the window would be "pushed down" so that it would fill the window if the user then scrolled forward. This complex back-scroll (") requires that we do our formatting backward or that we format the image several times until it comes out right. Both approaches would be costly. -- Charles.

1

CHI 5-MAR-75 23:59 25523

scrolling in nls

(J25523) 5-MAR-75 23:59;;;; Title: Author(s): Charles H. Irby/CHI;
Distribution: /JMB([INFO-ONLY]) WEC([INFO-ONLY]) RWW([
INFO-ONLY]) FEEDBACK([INFO-ONLY]) SGR([INFO-ONLY]) CKM([
INFO-ONLY]) ; Sub-Collections: SRI-ARC FEEDBACK; Clerk: CHI;

RLL 6-MAR-75 14:01 25524

Visit to Tymshare re: TYMNET [7 FEB 75

(DATE) 7 FEB 75 1

(BY) Lieberman 2

(ATTENDEES) 3

Paul Brickey of Tymshare 3a

Ed Pollack (EEP) of Tymshare 3b

Bob Martinez (BOBM) of Tymshare 3c

Jerry Wheat of Tymshare 3d

Art Case of Tymshare 3e

Jim Bair (JHB) of SRI-ARC 3f

Ray Panko (RA3Y) of SRI-ARC 3g

Robert Lieberman (RLL) of SRI-ARC 3h

(ADDRESSES) Full name of organization, address, and phone number 4

(MEDIUM) FACE-TO-FACE 5

(WHERE) Tymshare, Cupertino, CA 6

(ACTION-ITEMS) 7

None 7a

(DISTRIBUTION) DCE JCN RLL RA3Y JHB MEH ARC-LOG RWW 8

(REFERENCES) 9

(REMARKS) 10

The attendees from SRI-ARC went to Tymshare to find out about the current and future status of their communication network. 10a

Ed Pollack (our current contact point for the utility) set up the meeting. Paul Brickey was the salesperson who was to give the presentation. 10b

Paul was a stereotype salesperson and proceeded to give the usual sales talk. 10b1

We listened for awhile and then steered away from that level and ask more technical questions. 10b2

Paul was obviously not the man to pin down on this. Ed Pollack seemed to speak from some knowledge and Bob Martinez contributed. 10b3

Finally Paul went out and brought back Art Case who clearly was a member of the technical staff. He answered many of our questions and appeared knowledgeable. 10b4

These are my notes from the entire 2 hour talk. 10c

Hardware and software 10d

The BASE is the software in a host that connects to the net. 10d1

Tymnet has about 130 Varian 620L's called Tymsats and Tymcoms with access in 73 cities. 10d2

The Tymsats are the minicomputers that user terminals dail into and the Tymcoms are the minicomputers that hook up to the host computers. 10d2a

Their current Tymsats have 8K memory with 4k devoted to buffers. 10d2b

The Tymcom minicomputer is hooked into the host computer (the DC-10 line scanner for the PDP-10) and looks like a 103A or 113A to the computer. 10d2c

Thus the Tymnet service appears just like any other dialup line coming in. 10d2c1

The packets for the Tymnet are 64 bytes. They are internal and may mix several customers. 10d3

The baud between host and node is 4800 at the moment. 10d4

There is no option for the present Tymnet for echoing of characters. It is fixed with the specification of terminal type. 10d5

At present when the system is outputting, the echoing goes to deferred. This is the only switching of echoing for the current Tymnet. 10d5a

parity can be supplied on output to the user terminals. 10d6

Visit to Tymshare re: TYMNET [7 FEB 75

Tymcoms cannot talk to a Tymcom; hence computer to computer connections are not possible. 10d7

Stanford Universty has a PDP-10 KI10 on both Tymnet and ARPANET (The latter connection should happen soon.). 10d8

The SU connection is called SUMEX. Their hookup is different than most. 10d8a

It goes directly to the Tymnet via the memory bus. This needs special code on the PDP-10. 10d8b

The software will be in the public domain so that it most likely will be readily available. 10d8b1

The reason for directly going into the memory bus is for speed. 10d8c

There is apparently a multiplexor for the ports going into the memory bus. 10d8d

Currently only two hosts per node are allowed. 10d9

Right now the baud rate is 1200. It will go up to 9600 baud for the terminal speed. 10d10

Tymnet II 10e

Tymshare is now beginning to upgrade their minicomputers for the pending Tymnet II. 10e1

Both Varian 73 computers and Interdata equipment are on order. 10e2

The Varian 73's will allow 4 hosts to be attached and have 90 dialup and sixteen 9.6Kb ports. 10e3

In Tymnet II one can have either deferred or immediate echoing. This can be adjusted by the computer host and the user via the host. 10e4

The new Tymnet will allow 2780 RJE type terminals in 6 months (2000 baud) and 3270 type as well. 10e5

It is conceivable to program modules into the new supernode (Interdata mini) to handle cases like minicomputers (for example he front end). 10e6

The new Tymnet will have plenty of buffer space and that they

do not believe they will have any problems there, (my comment: overly optimistic). 10e7

The supervisor will continue to do most of the monitoring. 10e8

Tymnet II will have 50Kb lines from the present 9600 baud lines. Not all will be the 50kb variety though. 10e9

In many cases they will have several 9.6 lines rather than one large line between two nodes. 10e10

Tymnet II will have dynamic rerouting in 6 months according to the salesperson but the technical person suggested more like 12 to 18 months. 10e11

General 10f

They have two lease cables to London and Paris. 10f1

They also have a satellite backup and the special software to handle it. 10f2

They have plans to go to Hawaii but not in the very near future. 10f3

There is a 60 day wait for a full hookup and only a 90 day commitment. 10f4

They have run cassettes via the net with no buffer problems. 10f5

Big push this year is to increase terminal speed capability. 10f6

They gross $2 million per year for the Tymnet operation. 10f7

Scenario of a terminal session. 10g

dial up 10g1

type a character identifying the baud rate and padding characteristics 10g2

type your name, password 10g3

the network will let you know if a circuit has been made, usually within 5 seconds. 10g4

From now until you hang up the phone line, you are unaware of the network. There is no way to hail the network directly.

RLL 6-MAR-75 14:01 25524

Visit to Tymshare re: TYMNET [7 FEB 75

(J25524) 6-MAR-75 14:01;;; Title: Author(s): Robert N. Lieberman/RLL; Distribution: /DCE([INFO-ONLY]) JCN([INFO-ONLY]) RLL([INFO-ONLY]) RA3Y([INFO-ONLY]) JHB([INFO-ONLY]) MEH([INFO-ONLY]) ARC-LOG([INFO-ONLY]) RWW([INFO-ONLY]) ; Sub-Collections: ARC-LOG SRI-ARC; Clerk: RLL;

You must go through the host (assuming the host has the appropriate software to communicate to the network). 10g5

Reliabilty: 10h

one hardware problem per year on the current Tymsats. 10h1

Maintenance 10i

Tymshare handles all maintenance except phone lines but assumes the responsibility to get the phone line repaired. They have nearly a $2 millllion phone bill a year. 10i1

Some modems are maintained by Tymshare. 10i2

Costs for Tymnet I (only) 10j

The pricing of a 4800 baud connection will be available about 15 May 75. 10j1

The pricing for Tymnet II is unknown at the moment. 10j2

A. Each logon to host computer $.50/each 10j3

B. accumulative per month time connected to host for all terminals 10j4

0 to 500 hours 3.00/hour 10j4a

next 1500 hours 2.50 10j4b

next 3000 hours 2.00 10j4c

next 5000 hours 1.50 10j4d

each hour over 10,000 1.00 10j4e

C. Transmission of characters 0.125/1000 char 10j5

between users and host computer (i.e., both ways) 10j5a

D. TYMCOM-III rental (30 ports) 2150./month 10j6

To buy = $40k 10j6a

(includes line between TYMCOM and net, maintenance, accounting, etc.) 10j6b

E. One time engineering installation charge. 1000. 10j7

(DOCUMENTS) Hard copy given and received 11

(GIVEN) Date and documents given 11a

(RECEIVED) Date and documents received 11b

Information sheet, "The Tymshare Network", Tymshare, 1971 11b1

Information sheet, "Tymnet Users", Tymshare, no date 11b2

Information sheet, "Network Services Node Access Locations", Tymshare, May 1975 11b3

A map of the AT&T long lines for the Tymnet, January 1975 11b4

RLL 6-MAR-75 14:01 25524

Visit to Tymshare re: TYMNET [7 FEB 75

(J25524) 6-MAR-75 14:01;;; Title: Author(s): Robert N. Lieberman/RLL; Distribution: /DCE([INFO-ONLY]) JCN([INFO-ONLY]) RLL([INFO-ONLY]) RA3Y([INFO-ONLY]) JHB([INFO-ONLY]) MEH([INFO-ONLY]) ARC-LOG([INFO-ONLY]) RWW([INFO-ONLY]) ; Sub-Collections: ARC-LOG SRI-ARC; Clerk: RLL;

DVN 6-MAR-75 15:23 25525

Helpless at BB&N

Running through a TIP only to BBN-B <CTRL-Q> yeilds the message "Help command system Error, call ARC" whereas the command Help crashes you promptly into exec. I presume this is because the name <documentation> is prempted by a BBN File and the data base is in <arcdocumentation, help,>. If that is the case, couldn't we change the searching code? 1

DVN 6-MAR-75 15:23 25525

Helpless at BB&N

(J25525) 6-MAR-75 15:23;;;; Title: Author(s): Dirk H. Van Nouhuys/DVN; Distribution: /FEED([ACTION]) HGL([ACTION]) DIRT([INFO-ONLY]) SGR([INFO-ONLY] if you want to show help to Warshall, you will have to go o office-1) ; Sub-Collections: SRI-ARC DIRT; Clerk: DVN;

POOH 6-MAR-75 19:06 25526

changing the telnet escape character

I highly recommend always changing your telnet escape character to <CTRL-Y) when you have to change it. It would help eliminate some of the confusion when leaving a terminal without completely logging out...the elf queen 1

POOH 6-MAR-75 19:06 25526

changing the telnet escape character

(J25526) 6-MAR-75 19:06;;;; Title: Author(s): Ann Weinberg/POOH;
Distribution: /SRI-ARC([ACTION]) ; Sub-Collections: SRI-ARC; Clerk:
POOH;

The Evolving Ever-Changing Emlf Userguide

This version of the ELF Userguide is updated as of Friday February 28. It is in the process of being revised and additions and corrections are glady accepted. A new version will be out.....the elf queen

The Evolving Ever-Changing Emlf Userguide

Introduction 1

Because the new ELF System supports more applications than the old system, using it for network access is slightly more complicated. The command language is like the TENEX executive. This language supports logging in, performing various utility functions, running programs, and logging out. The portion of ELF which interprets user commands is referred to as the ELF executive, or the ELF EXEC. Utility functions include finding out who is using the system and sending messages to one or all other terminals. At present the only program of general interest is TELNET. It is this subsystem that performs the network access function (the total function of the old system). In the future, subsystems will be added to support such things as the full FTP protocol (in much the same way TENEX does) and access to local resources. 1a

This document introduces the present ELF system while more features are being added and more complete documentation is being prepared. 1b

Notation 2

In ELF, as with TENEX and other systems, commandwords can usually be identified by the first few letters. In this document, commandwords are capitalized e.g. Login. Each commandword is spelled out completely with the optional letters in square brackets. As an example, the Login commandword is used to log into the system but only the first three letters are required: Log[in]. 2a

Following the commandword specifying the operation there may be zero or more operands, some of which may be optional. In this document the operand description is all uppercase, and is enclosed in square brackets if it is optional. For example the Login keyword should be followed by two operands. The first of these specifies the user's name. This name indicates who is using the terminal when other users request the Systat operation. For the second operand, the user may specify a password or type a space. The following notation is an example: 2b

```
Log[in] <SP> USERNAME <SP> [PASSWORD] <CR>
```
2b1

A space (represented by <SP>), and an escape, or an alt-mode (represented by <ESC>) terminates a field. A carriage return (represented by <CR>) terminates a field and the line. 2c

Executive Commands 3

Following is a list of executive commands together with a brief

description of each. They are arranged in alphabetical order for easy reference. You might want to read the descriptions for the first time in this order: 3a

?, Login, Systat, Version, Run, Continue, Logout, Detach, Attach, Noecho, Echo, Reset, Enable 3a1

The following commands are only available to enabled users: Disable, Drain, Release, 3a2

?
This command lists the alternatives available to you. 3b

Att[ach] <Sp> JOBNUMBER <SP> [PASSWORD] <CR>
The terminal is attached to the specified job if the following conditions are met:
(1) The job specified is currently detached.
(2) The password matches the password of the job specified (unless there is no password associated with the job). 3c

Con[tinue] <CR>
The previously running subsystem continues from where it left off when it was last interrupted by a <CTRL-C>. 3d

Det[ach] <CR>
The terminal is detached from the active job. At a later time the Attach command can be used to attach the same or a different terminal to the job. 3e

Dis[able] <CR>
The command disables the job so that enabled commands are no longer permitted (see Enable, below). 3f

Dr[ain] <CR>
No further logins are permitted. This command can only be used if the job is enabled (see Enable, below). 3g

Ec[ho] <CR>
The command specifies that the system should echo each character as it is typed. As the echo state is the default, this command need only be used when you have previously specified Noecho (see below). 3h

Ena[ble] <CR>
This command is used to enable the job. After the job is enabled certain other commands can be used. The herald is changed to an exclamation point (!). 3i

Log[in] <SP> USERNAME <SP> [PASSWORD] <CR>

you login. After being logged in the set of commands described here are available to you. Some of the commands listed are only available if your job is enabled. 3j

Logo[ut] [<SP> JOBNUMBER] <CR>
You logout. If no job number is specified the active job is logged out. The job number cannot be specified unless the active job is enabled. 3k

Noe[cho] <CR>
This command specifies that no further echoing should be performed by the system. This would be used with certain terminals (half duplex) that do their own echoing. 3l

Rel[ease] <CR>
This command is the opposite of the Drain command. It returns the system to the state where new jobs will be logged in as long as there are sufficient resources. This command can only be used if the job is enabled. (See Enable above.) 3m

Res[et] <CR>
This command closes all open files and deletes the active subsystem (if one exists) making it impossible to Continue. This function is performed automatically when you run another subsystem or log out. 3n

Ru[n] <SP> SUBSYSTEMNAME <CR>
The specified subsystem is run if there are sufficient resources. To return to the executive command mode you should type <CTRL-C>. Note that if you just specify a subsystem, the Run command is assumed. (currently only TELNET) 3o

Sy[stat] <CR>
The ELF system statistics are given including a list of the jobs currently logged into the PDP-11. 3p

Tel[net] <CR>
A subsystem of ELF (not an ELF command, see below) 3q

V[ersion] <CR>
The current ELF system version number is given. 3r

The TELNET Subsystem 4

This subsystem provides access to the ARPA NETWORK. A connection can be established with a foreign host computer and later closed. The commands listed below are currently available. In the following discussion, |or| denotes a choice between two items. Noise words are in parenthesis. 4a

Co[nnect] <SP> (to) [HOSTNAMES !or! HOSTADDRESS] [SOCKETNUMBER] <CR>

This command requests a connection with a specified host. If no host is specified the previous connection (if there was one) is resumed. Only certain hosts can be specified by name. All other hosts must be specified by their numeric address (normally in octal - if you wish to use decimal, prefix the number with the two characters "^" and "D"). Following is the beginning of a growing list of the hosts that can be specified by name from ARC.

B[BN-TENEXB]
O[FFICE-1]
SR[I-AI]
N[IC]
I[SI-KI-TENEX]
M[IT-MULTICS]

At other sites there may be a different set of hosts that can be specified by name. 4b

Note that if no command is specified, the subsystem assumes that a connect was requested. For example, all of the following request a connection to BBN-TENEXB. 4c

Connect <SP> BBN-TENEXB <CR> 4c1

Co <SP> B <CR> 4c2

B <SP> <CR> 4c3

Cl[ose] [<SP> HOSTNAME !or! <SP> HOSTADDRESS] [<SP> SOCKETNUMBER] <CR>

The specified connection is closed. If no operands are specified, all open connections are closed. Since at present only one connection can be open at a time, Cl <CR> is sufficient to close the currently open connection. 4d

D[isconnect] (from) [<SP> HOSTNAME !or! <SP> HOSTADDRESS] [<SP> SOCKETNUMBER] <CR>

This command is exactly the same as the Close command. 4e

S[tatus] [<SP> HOSTNAME !or! <SP> HOSTADDRESS] [<SP> SOCKETNUMBER] <CR>

This command determines the status of a particular host or the local IMP. At present, you can only learn the status of the local IMP. S <CR> will give you the local IMP's status. 4f

E[scape] <ESC> (=) ANYCHARACTER <CR>

This command changes the default escape character which is <CTRL-Z>. The escape character is used to return to TELNET command mode after a connection has been established. To return

to the connection after using the escape character you need only type <CR>. Reset the escape character to <CTRL-Z> using this same command. 4g

De[Bug] <SP> on !or! off <CR>
This command turns the NCP debug option ON and OFF. It is only available to jobs that are enabled in ELF. 4h

Q[uit] <CR>
This command is used to quit TELNET and return to the executive. 4i

LOGIN and LOGOUT Procedures from SRI-ARC through ELF to BBN-TENEXB
Things surrounded by exclamation points are comments. You type everything that is not surrounded by exclamation points. 5

!Turn on the terminal and the lineprocessor! 5a

<CTRL-C>
!brings you the ELF @! 5b

Log <SP> USERNAME <SP> PASSWORD <CR>
!log in with name and password to ELF! 5c

Tel <CR>
!runs the TELNET subsystem! 5d

B <CR>
!connects you to BBN-TENEXB(SRI-ARC) and brings you the TENEX login message! 5e

Log <SP> USERNAME <SP> PASSWORD <SP> <CR>
!log in with name and password at BBN-TENEXB! 5f

!The next step is necessary only if your characters are not echoed when you begin to login.!
<CTRL-C> Full <CR> 5g

!The next step is necessary only if you are using a lineprocessor.!
Ter <ESC> Li <ESC> <CR> 5h

No <SP> Rai <CR>
!gives you upper and lower case characters! 5i

!At this point you can login to NLS! 5j

Logo <CR>
!log out from BBN-TENEXB! 5k

<CTRL-Z>
!return to TELNET! 5l

Q <CR>
!return to local ELF executive! 5m

Logo <CR>
!log out from local ELF system! 5n

LOGIN and LOGOUT Procedures from SRI-ARC through the AI-SRI PDP-10 to BBN-TENEXB
Things surrounded by exclamation points are comments. You type everything that is not surrounded by exclamation points.. 6

!Turn on the terminal and the lineprocessor! 6a

<CTRL-C>
!brings you the AI-SRI @! 6b

SRI-ARC <SP> NEXTDOOR <SP> <CR>
!log in with this name and password to AI-SRI! 6c

Telnet <CR>
!runs TELNET! 6d

Tran <CR>
!puts the TIP in the transparent mode so all that characters you type are sent to the host you intend to talk to! 6e

Conn <SP> BBNB <CR>
!connects you to BBN-TENEXB(SRI-ARC) and brings you the TENEX login message! 6f

Log <SP> USERNAME <SP> PASSWORD <SP> <CR>
!log in with name and password at BBN-TENEXB! 6g

!The next step is necessary only if your characters are not echoed when you begin to login.!
<CTRL-C> Full <CR> 6h

!The next step is necessary only if you are using a lineprocessor.!
Ter <SP> Li <SP> <CR> 6i

No <SP> Rai <CR>
!gives you upper and lower case characters! 6j

!At this point you can login to NLS! 6k

Int <ESC> <CR>
!removes you from the transparent mode! 6l

Logo <CR>
!log out from BBN-TENEXB! 6m

<CTRL-Z>
!return to TELNET! 6n

Gui <CR>
!return to AI-SRI! 6o

LOGO <CR>
!log out from AI-SRI PDP-10! 6p

Tentative Printing Procedures 7

NOTE: Output Quickprint will not work (unless you have friends at BBN who will mail you your copies). 7a

Files must be sequential in order to be printed 7b

Use the Output Sequential or Output Terminal File command on an NLS file 7c

Goto Tenex 7d

Use the command: Copy FILENAME (to) NET:163-10 <CR>
The system should respond with ASCII
Type <CR> <CR> and 7e

The information below pertains to the original ELF USER'S Guide. Changes were made to reflect the use of ELF at SRI-ARC and SRI-ARC notation conventions. 8

January 1, 1975 8a

B. W. Schafer 8a1

D. L. Retz 8a2

J. R. Miller 8a3

J. L. McClurg 8a4

Speech Communications Research Laboratory, Inc. 8b

800-A Miramonte Drive 8b1

Santa Barbara, California 93109 8b2

(805) 965-3011 8b2a

This work is supported by the Advanced Research 8c

Projects Agency through Contract No. N00014-73-C-0221, 8d

administered by the Office of Naval Research. 8e

TITLE PAGE 9

THE EVOLVING EVER-CHANGING ELF USERGUIDEAu9mentation Research
CenterStanford Research Institute
333 Ravenswood Avenue
Menlo Park, California 94025 9a

POOH 6-MAR-75 19:21 25527

The Evolving Ever-Changing Emlf Userguide

(J25527) 6-MAR-75 19:21;;; Title: Author(s): Ann Weinberg/POOH;
Distribution: /SRI-ARC([INFO-ONLY]) ; Sub-Collections: SRI-ARC;
Clerk: POOH; Origin: < WEINBERG, ELFUSE,NLS;22, >, 28-FEB-75
10:57 POOH ;;;; ####;

File Conversion Notes

This is a rough draft of a rough set of working notes. Questions and Comments are requested. 1

File Conversion Notes

(J25528) 7-MAR-75 05:12;;;; Title: Author(s): David S. Maynard/DSM; Distribution: /JBP([INFO-ONLY]) ; Sub-Collections: SRI-ARC; Clerk: DSM;

DVN 7-MAR-75 12:24 25529

Message Subsystem Broken at BBN

The message subsystem is not nworking at BBNB. It loads but when you give the command Move Message, it sends you into exec with "ILLEGAL INSTRUCTION JSYS 63 = 104000,,63 at PROPRO+7 etc etc." This is playing havokc with various people's communication with the world. 1

DVN 7-MAR-75 12:24 25529

Message Subsystem Broken at BBN

(J25529) 7-MAR-75 12:24;;;; Title: Author(s): Dirk H. Van Nouhuys/DVN; Distribution: /FEED([ACTION]) RWW([INFO-ONLY]) DCE([INFO-ONLY]) KIRK([INFO-ONLY]) NDM([INFO-ONLY]) ;
Sub-Collections: SRI-ARC; Clerk: DVN;

DvN 7-MAR-75 12:54 25530

Limited Facilities for User Interaction With Command Branches

This responds to a sendmessage from Jeanne Beck to me expressing Connie McLindon's interest in more powerful commands branches.

DVN 7-MAR-75 12:54 25530

Limited Facilities for User Interaction With Command Branches

Facilities exist for users interacting with command branches only in a very limited way. A procedure replacement exists in my directory at BBN or in Pat Whitting-Okeefe's directory at office-1 that makes commands branches stop and wait for input from users. To use it you Goto Program and load the program by its name (auxchr,). To use it you insert percent signs (%) in the place in your command branch where you want the stream to stop. Each % in the branch waits for the user to enter one character. That is the catch. You must either know how many characters the user is going to need to put in, or put in a lot of %'s and have a way for her to put in harmless characters after her significant input is over. We see that it would be very useful to add more flexible user input and IF constructions to commands branches. It is merely a matter of shaking loose programming time to do it. 1

DVN 7-MAR-75 12:54 25530

Limited Facilities for User Interaction With Command Branches

(J25530) 7-MAR-75 12:54;;;; Title: Author(s): Dirk H. Van Nouhuys/DVN; Distribution: /FEED([ACTION]) JMB([ACTION]) CKM([INFO-ONLY]) JHB([INFO-ONLY]) JCN([INFO-ONLY]) RWW([INFO-ONLY]) KEV([INFO-ONLY]) KIRK([INFO-ONLY]) DCE([INFO-ONLY]) PWO([INFO-ONLY]) ; Sub-Collections: SRI-ARC; Clerk: DVN;

CHI 5-MAR-75 20:45 25531

Insert date command

In reply to your message of 2-MAR-75 20:21 Susan/FEED
Journal: (31941,)
Subject: acknowledgement of (25449,) Insert date command

1

Date need not be a second level commandword in the Insert Date command. JDH should change this for all NLS's in the World. CHI 1a

CHI 5-MAR-75 20:45 25531

Insert date command

(J25531) 5-MAR-75 20:45;;; Title: Author(s): Charles H. Irby/CHI;
Distribution: /FEED([INFO-ONLY]) JDH([INFO-ONLY]) ;
Sub-Collections: SRI-ARC; Clerk: CHI;

KEV 7-MAR-75 17:22 25532

dropped line processor characters

just in case you havent noticed we are experiencing problems with dropping characters between lineprocessor terminals and bbn. until we solve these buffer problems you can aid in maintaining your own sanity (if you wish to) by using the following alternative input techniques: 1

use command accept, command delete, backspace character and/or word, etc. from the keyboard instead of the mouse buttons 1a

use the set viewspecs command rather than entering viewspecs with the mouse buttons down 1b

in general, if something can be done from the keyboard or the mouse buttons, the use of the keyboard will cause less characters to be lost 1c

note that these alternatives are purely personal options and in no way affect overall system response and that they will not solve the problem entirely but may make it less annoying 2

KEV 7-MAR-75 17:22 25532

dropped line processor characters

(J25532) 7-MAR-75 17:22;;; Title: Author(s): Kenneth E. (Ken) Victor/KEV; Distribution: /SRI-ARC([INFO-ONLY]) ; Sub-Collections: SRI-ARC; Clerk: KEV;

CHI 7-MAR-75 18:25 25533

NLS at BBNB

Have either of you compared the responsiveness of DNLS at BBNB (8 network hops from ARPA TIP) to that of DNLS at Office-1 (13 hops)? Would be very interesting data to us. -- Charles. 1

CHI 7-MAR-75 18:25 25533

NLS at BBNB

(J25533) 7-MAR-75 18:25;;;; Title: Author(s): Charles H. Irby/CHI;
Distribution: /SGR([INFO-ONLY]) JMB([INFO-ONLY]) ;
Sub-Collections: SRI-ARC; Clerk: CHI;

JBP 9-MAR-75 23:36 25535

pcp questions from dmw of multics

9-MAR-75 18:28:23-EDT,1435;000000000000
Mail from MIT-ML rcvd at 9-MAR-75 1828-EDT
Date: 8 MAR 1975 2346-EDT
From: DMW at MIT-ML
To: postel at BBN-TENEXB, dmw at MIT-MULTICS

More questions:

1) In as much as we can't easily convince our TBH supervisor to multiplex itself, would we cause much problem if we required the WM to create a subprocess to run the file-package. This would presumably leave the TBH supervisor as only required to support PMP (plus PSP, etc.).

2) In PCP there is an interrupt mechanism. What should either/both ends of PCP do if an INTPRO and a RTNPRO cross in mid-network such that one end of PCP receives an INTPRO for which it finds it has no active procedure, and the other end receives a permanent return (or any of the other possible temporary returns) for which it is expecting an INTERRUPTED return. As I see it, only if you specify what is expected of PCP implementers will you get a consistent set of implementations. (As I see it, this should not be interpreted as a PCPERR type of problem.)

3) We seem to now have an implementation schedule which calls for an implementation of FP starting about April 1. Do you know what the status of the WM's requirements for the generic procedures <open file>, <close file>, and <warrant file> are? That is, are Millstein, et ali., making any progress in defining the interface they wish?

- Doug Wells

JBP 9-MAR-75 23:36 25535

pcp questions from dmw of multics

(J25535) 9-MAR-75 23:36;;; Title: Author(s): Jonathan B. Postel/JBP; Distribution: /JEW([INFO-ONLY]) ; Sub-Collections: SRI-ARC; Clerk: JBP;

RLL 10-MAR-75 00:24 25536

bug: process command not working at bbnb

th e process comand does not seem to work at bbnb. please let me know when it gets fixed. i aave a demo this week and it is a very nie feature that i usual ly show,. thanks. here is the error messge i get , illegal instruction jsys sqsyw+107 = 104000,,521 at auxchr+2i = 135221 theen the usual register stuff which i wont bother to print here. robert

1

RLL 10-MAR-75 00:24 25536

bug: process command not working at bbnb

(J25536) 10-MAR-75 00:24;;;; Title: Author(s): Robert N. Lieberman/RLL; Distribution: /FEED([ACTION]) JDH([ACTION]) ; Sub-Collections: SRI-ARC; Clerk: RLL;

RLL 10-MAR-75 00:26 25537

bug: move message does not work at bbnb

the message subsystwm command m mmve messge does not seem to work at bbnb, let mej know the staus of fising it there, thanks, robert 1

RLL 10-MAR-75 00:26 25537

bug: move message does not work at bbnb

(J25537) 10-MAR-75 00:26;;;; Title: Author(s): Robert N. Lieberman/RLL; Distribution: /FEED([ACTION]) JDH([ACTION]) ; Sub-Collections: SRI-ARC; Clerk: RLL;