Proposal for Research No. ISU 74-132
Development Support NSW

(J23945) 12-SEP-74 13:32;;;; Title: Author(s): Richard W. Watson/RWW; Sub-Collections: SRI-ARC; Clerk: JOAN; Origin: < WATSON, NEWNSW.NLS;51, >, 7-AUG-74 09:26 RWW ;;;;(WATSON, NEWNSW.NLS;50,), 16-JUL-74 15:31 RWW ;

(note: When journalizing, use this preassigned number - assigned to MEJ) ####;

Proposal for Research No. ISU 74-132
Development Support NSW

SUBJECT:

NATIONAL SOFTWARE WORKS DEVELOPMENTS

1

SUMMARY: 2

This proposal by the Augmentation Research Center (ARC) of SRI covers tasks for the National Software Works Program (NSW). 2a

The National Software Works Program has as its goal the creation of an ARPANET based resource sharing framework into which tools can be embedded that aid each phase of the software system development process. Its framework is designed to promote the easy integration and coordination of new tools and the gradual strengthening of their cooperation into a unified system [1,2,3]. (References are listed at the end of Part One of this proposal.) 2b

The tasks proposed here are of two main types: tasks that assist in the creation of the NSW ARPANET-based system framework, and tasks that provide initial tools for the NSW environment. 2c

There are four main components of the NSW system: 2d

1) An NSW Frontend system (frontend) that will provide terminal access to the ARPANET and provide a set of services that will create a coherent NSW user environment. 2d1

2) An NSW Works Manager that will provide special services such as authentication, record keeping, file system, resource interconnection and file transfers, management aids, and so forth. 2d2

3) Protocols and conventions needed between the frontend and Works Manager, frontend and tools, and Works Manager and tools. 2d3

4) The tools (Computers and software) that will reside in the NSW environment. 2d4

ARC is proposing the following tasks: 2e

System Framework Tasks 2e1

1) Design and implementation of the NSW Frontend system. 2e1a

2) prime responsibility for the management of the development of protocols needed by the NSW system. It is assumed that collaboration will be required with other NSW contractors, and other ARPANET communities with similar protocol needs. Other NSW contractors are tasked to develop some of the required protocols. 2e1b

Tool Tasks 2e2

3) Enhancements to ARC's oNLine System (NLS) to make it an even more effective tool for the NSW environment. 2e2a

4) Software for a DEC PDP-11 needed to connect the Air Force Burroughs 3500 Computer at the Air Force Data Systems Design Center in Montgomery, Alabama, to the ARPANET. 2e2b

This latter task is proposed as optional and is costed separately in Part Two - Contractual Provisions. 2e2b1

Proposal for Research No, ISU 74-132
Development Support NSW

OBJECTIVES: 3

I. Design, implement, and document a new NSW Frontend System. This system is to function with the other parts of the total NSW system [2,3]. 3a

A DEC PDP-11 based frontend satellite computer system will be developed for use with the National Software Works software production facility. This system will generally sit between the user's terminal and the ARPA network, but can also be accessed from terminals connected to TIPs. A version of the system will also run on a PDP-10 TENEX system as well. This frontend system will interact with the user who is giving commands to the NSW Works Manager, or to some tool operating within the NSW environment. It will provide terminal control, aid the user in command specification, parse commands, and communicate with the appropriate resource(s). It will play an important role in providing the coherent user and unified system environment desired in the NSW system. 3a1

II. Take responsibility for the design, documentation and follow through of those protocols and conventions required to create the ARPANET resource sharing environment needed by the NSW system. 3b

In the past, responsibility for protocol development in the ARPANET environment was widely shared and on a voluntary basis. This approach, while having achieved a number of successes, has had some serious deficiencies due to the lack of prime responsibility and accountability. To achieve the goals of the NSW, many new protocols and conventions need to be developed (as described later), existing protocols need to be reexamined and possibly redesigned, and management control is needed in the form of scheduling, obtaining protocol approval, consulting, validation, and other tasks associated with protocol implementation. 3b1

III. Provide restructuring and enhancements to NLS to make it operate most effectively in the NSW environment. NLS will provide documentation, source code editing, and other services to the NSW user community. 3c

NLS will require some restructuring to operate with the NSW frontend and Works Manager in a fully integrated way. NLS is envisioned as the prime tool for documentation and source code creation within the NSW. NLS can provide

additional services as well, since it is in itself a collection of tools. To meet the needs of the initial NSW user community of COBOL system developers, a number of enhancements are proposed to increase its effectiveness in the NSW environment. Providing these special NLS enhancements is of prime importance to help COBOL programmers prepare and debug their source code files, to help both programmers and technical writers document the systems developed, and to assist clerical personnel in the input of material. A list of desirable modifications appears below in the discussion of the proposed work. 3c1

IV. (Optional Objective) Design, implement and document a DEC PDP-11 based software system to connect a Burroughs 3500 computer to the ARPANET in a manner requiring minimum modification to the 3500 operating system and extendable to other types of computers in the future. 3d

One of the initial tools to be placed in the NSW environment is a Burroughs 3500 computer at the Air Force Data Systems Design Center in Montgomery, Alabama. This computer is to be the target machine for software developed by the initial NSW user community. It is primarily a batch machine specialized for COBOL program development. Initial NSW studies concluded that this class of machine is most effectively attached to the ARPANET through a minicomputer. 3d1

Proposal for Research No. ISU 74-132
Development Support NSW

BACKGROUND AND TECHNICAL NEED: 4

The NSW Problem 4a

The goal of the ARPA NSW Program is to improve the productivity of the DoD software system building process by making available to production DOD system building staffs (programmers, documenters, managers, others) in a coordinated fashion the types of advanced tools in fairly widespread use in the research community. The Air Force alone spends over one billion dollars a year on software developments. It is well known that software is unnecessarily costly [4]. One way to decrease the cost of the software development process is to make improved tools and methodologies available to software engineers, managers, documenters, and other people involved with the system building process [4]. An approach to making tools more widely available is to create a network based marketplace, providing a framework for commercial, government, and R&D organization developed products. The NSW effort aims to create such a marketplace and, while requiring new technology, is also seen as having a heavy technology transfer component. 4a1

The technical need is to provide a system framework within which classes of advanced and existing tools can intercommunicate within the ARPANET environment, to provide a coherent user environment that, to the extent possible, gives the user the feeling that he is dealing with a single unified system, and to choose and modify existing tools to better serve the production DOD programming community, initially chosen as those using COBOL as the implementation language. 4a2

There will be three classes of tools that must be accommodated within the NSW environment: existing tools that are not to be modified, tools that will operate within the conventions of the NSW environment and outside the NSW environment, and tools that will operate strictly within the NSW environment. 4a2a

Past Efforts 4b

Many tools have been developed to aid aspects of the software system building process. These tools are not, however, widely available to DoD software developers because they are usually restricted for use on one computer system, or exist within the R&D community and are not

readily available in a form or environment compatible with the DOD software production process, Further, the tools are often disjointed, and have serious incompatibilities and conflicting user conventions, 4b1

There are many cases of DOD projects investing heavily in tool development before launching on a project only to have these tools lost after the development effort or system life was complete, There is considerable experience to indicate that small teams of skilled software developers, well supported with software system building, documentation, and management methodology and tools, are superior to large, mediocre, and poorly supported groups [4], 4b2

The advent of the ARPANET with its improvements in communication economies, and ability to enable communication and resource sharing between different manufacturers' and developers' equipment and software offers a significant new environment within which to tackle the above needs, 4b3

Reference 4 is an excellent overview of the present state of the software system building process and contains a number of main recommendations for research and development that are directly achievable either by the NSW system approach or within the environment that will be created by the NSW system, 4b4

As indicated below, ARC has extensive experience in working to develop a prototype coherent tool workshop for software engineering, ARC has focused heavily on this application area both because of its importance in its own right and as a vehicle to allow its system developers to use and test the concepts and system they have been developing, 4b5

ACCOMPLISHMENTS: 5

The Augmentation Research Center at SRI has been actively pursuing research and development in areas central to the NSW program since 1962. Relevant recent references are contained in [6-16]. ARC has made many significant accomplishments during this period. These accomplishments, coupled with a staff highly experienced in the building, application, analysis, and personal use of NSW type technology, provide the basis for the important roles ARC is proposing to play in the NSW program. A list of some of the accomplishments follows: 5a

1) ARC has a history of quality software engineering and has been a leader in applying new software engineering tools to aid the system building process. Examples follow: 5b

a) Early development of a meta-compiler system and its extensive use in building high level system programming languages. 5b1

b) Use of high level system programming languages in system building. 5b2

c) Pioneered early use of the ARPANET resource sharing to convert from an XDS 940 to a DEC PDP-10 computer. 5b3

d) One of the first groups to recognize the merits of TENEX and to adopt it and pioneer its early use. 5b4

e) Extensive use of NLS text features for source code preparation, documentation, file and procedure linking, and maintenance. 5b5

f) Use of split screens and source language debugging facilities. 5b6

g) Development of a Command Meta-Language for high level specification of the user interface. 5b7

h) Use of modular design principles. 5b8

i) Pioneered, with Xerox-Parc, development of a new modular runtime linkable programming system, MPS, to be released in summer '74. 5b9

j) Introduction and use of extensive measurement facilities in TENEX and NLS. 5b10

2) A strong early lead in getting collaboration going on TELNET, File Transfer, Mail, Graphics, and other ARPANET protocols. One of the first users of the ARPANET in a resource sharing mode for software development and a developer of the Host-Host protocol. 5c

3) Early explicit recognition of the potential that online computer and communication technologies have (in areas outside of straight numeric or accounting computation) in enhancing the effectiveness and efficiency of managers, scientists, engineers, programmers, and their supporting staffs in their daily work. 5d

4) Early explicit recognition of the importance to system building of an integrated system of text handling and system building tools. 5e

5) Publication of over 25 reports and papers on NSW related topics. 5f

6) Demonstration--at large professional meetings (FJCC '68, ASIS '69, ICCC '72), to hundreds of visitors, and via film--of a working prototype system containing many core concepts and features of an NSW system. The FJCC '68 conference was the first to show the power of coupled screens, video terminals, multiple display windows, and multimedia techniques (computer output, video pictures and a voice link). 5g

7) Pioneered the two-dimensional text work that will play an increasingly important role in the NSW environment, and developed many highly interactive tools and concepts for working and browsing in an information space, such as view specifications, interfile links, split screens, cross file editing, integration of text and numeric computation. 5h

8) Pioneered input device and work-station design (video displays, mouse, keyset, desk, and workspace). 5i

9) Long experience in high quality formatted publication quality hardcopy, through line printers, typewriters, and COM. 5j

10) Pioneered the concept of an integrated coherent workshop of many office tools with a uniform user interface. 5k

11) Early experience in teleconferencing support between remote individuals and groups at coupled screens (possibly video projected). 5l

12) Considerable experience with online information management for an office or software project environment, such as memos and correspondence, full text storage and retrieval, indexing, and cross linking. 5m

13) First with a comprehensive system for online message control, addressing, distribution, delivery, individual and group identification, cross linking, and indexing. 5n

14) Over one hundred thousand hours of hands-on console experience with the use of NSW type technology in daily work, both at ARC and out on the ARPANET. 5o

15) Recognition of the importance of integrating mechanisms for studying and facilitating technology transfer into the system building process. 5p

a) Early application experience with the Network Information Center. 5p1

b) Pioneered the establishment of a solid service with the NLS Tymshare operated Utility (OFFICE-1) for support of applications. 5p2

c) Establishment of training and other application support services. 5p3

PROPOSED EFFORT: 6

NSW Frontend Development 6a

The core concepts in the NSW frontend will be based on the experience gained in the development at ARC of a Command Meta-Language (CML) System for NLS. The NSW CML system will, however, allow greater flexibility both in terms of the range of command languages that can be expressed in CML, and also the extent of individual tailoring and usage measurement that the system will allow. 6a1

Features of the frontend are discussed below: 6a2

User Features 6a2a

The NSW frontend will allow the user access to all of the facilities provided in the NSW. All commands to the Works Manager or to any tool must be given through the frontend. The frontend will provide the user with a range of tailorable command language disciplines. For example, it will allow the user to specify how much prompting he would like when using each tool. In addition, the user may request terse syntactic/semantic help when specifying commands or verbose conceptual help with any tool or the NSW system facilities. 6a2a1

The frontend will create a coherent user environment for the workshop of tools that the NSW environment will provide. It will provide a common set of principles and conventions for the various user interface features, command language, control conventions, and methods for obtaining help and computer aided assistance. 6a2a2

Creating a coherent user environment has two main implications. One, it means that while each tool domain within the NSW may have a vocabulary unique to its area, this vocabulary will be used within language and control structures common throughout the workshop. A user will learn to use additional functions by increasing vocabulary, not by having to learn separate "foreign" languages. Two, when in trouble, he will invoke help or tutorial functions in a standard way. 6a2a3

The frontend will provide a set of capabilities to accommodate both novice and expert users. 6a2a4

System Level Features 6a2b

Computer Hardware Selection 6a2b1

Hardware discussions to date indicate that the decision has been made to use an appropriately configured DEC PDP-11 system as the NSW frontend machine. ARC will aid in specifying the appropriate configuration. 6a2b1a

Multiuser Operating System 6a2b2

Alternative operating systems will be evaluated in conjunction with other NSW contractors, and a decision will be reached on which to use. 6a2b2a

ARPANET Protocols 6a2b3

The standard ARPANET NCP Program will provide primitives for communicating with other NSW processes over the ARPA Network and should be included in the operating system. Other ARPANET protocols needed within the NSW environment will also be provided. Some of these protocol support processes will be provided with the operating system. Others must be developed. In particular, protocols must be designed and implemented for communication between the frontend and the Works Manager and NSW tools. These conventions must enable communication of commands and arguments from the frontend to the Works Manager and tools. The conventions must also enable tools or Works Manager to tell the frontend to prompt the user for additional information and describe changes in the displayed image. Frontend protocol development must be managed within the framework of total NSW protocol development. (See section on Protocols Development below). 6a2b3a

Operating System Interface 6a2b4

An operating system interface module will be implemented, consisting of a set of virtual operating system primitives. Use of these primitives will be converted to actual operating

system calls. The goal here is to make the frontend as operating system independent as possible. This will enable the frontend to reside on both a PDP-11 and PDP-10 or be transferred to other equipment later. 6a2b4a

Command Language Interpreter 6a2b5

The Command Language Interpreter (CLI) will interact with the user to allow him to give commands to various NSW tools, the Works Manager, or other NSW facilities. The CLI must also be able to handle machine-oriented messages from NSW tools or the Works Manager and translate these messages to an appropriate man-oriented language form. The interpreter will run on either a DEC PDP-11 or PDP-10 (TENEX). 6a2b5a

It may also, depending on overall NSW system design, inform the Works Manager whenever the user attempts to access or create new files, save the user's commands for intervals between tool checkpoints (if the tool is capable of checkpointing its environment), allow the user to reissue old commands, allow the user to define and use command macros (abreviations) and interact with tools (upon user request) to "undo" the effect of commands where possible. 6a2b5b

The Command Language Interpreter is driven by two data structures, a Command Language Grammar and a User-Profile data structure, described below. These data structures are sent to the frontend either by the Works Manager or by the tool system as needed. The command language available to the user at a point in time is represented by the union of the grammars active at that point. 6a2b5c

Command Meta-Language 6a2b6

A Command Meta-Language (CML) for specifying the user interface will be developed. A compiler will be implemented to produce command language grammar data structures from the CML specifications. 6a2b6a

Command Language Grammars 6a2b7

A Command Language Grammar is a data structure that describes the command language for a tool or the NSW Works Manager. When the user changes tools, new grammars will be loaded from the Works Manager or tool, if necessary. 6a2b7a

User-Profile 6a2b8

A User-Profile is a data structure that is loaded by the frontend when the user is authenticated by the Works Manager. It may be unique to each individual user and it is used by the Command Language Interpreter to determine how it should interact with that user. This data structure controls such things as how much help or prompting a user receives when using a particular tool, what commands are available to this user, and other information tailoring the system to the user. The information in this data structure can be changed upon user request or adaptively by programs based on user behavior. Updates to the User-Profile must be reported to the Works Manager. 6a2b8a

Terminal Control 6a2b9

This component interfaces the Command Language Interpreter to the particular display or typewriter terminal being used. Displays can be used as typewriter terminals or as full two-dimensional devices. The Terminal Control contains primitives for subdividing display screens into rectangular windows and for allowing the user to select text displayed on a screen as arguments in commands to tools. 6a2b9a

At the time this proposal is being written, we do not know whether or not NSW displays have been selected for the initial users. If not, we would participate in their selection. 6a2b9b

Because certain classes of work that the users want to do within the NSW environment involve remote job entry to batch processing systems, it is also desirable that the frontend be able to handle devices, such as card readers and punches,

line printers, and tape drives, as well as interactive terminals. 6a2b9c

Maintenance 6a2b10

The frontend systems will be maintainable, loadable, and (symbolically) debuggable from a remote timesharing (TENEX) system. The main features needed to support remote debugging are expected to be supplied with the operating system. 6a2b10a

Integration 6a2b11

The frontend system will be integrated into the complete NSW system. 6a2b11a

Documentation 6a2b12

The documentation to be provided will consist of: (1) an online user HELP Facility and command reference manual, and (2) system overview descriptions and well-commented source code listings. 6a2b12a

Protocol Development and Management 6b

As should be done for any large system built by many individuals or groups, the NSW will be designed and implemented as a set of interconnected program modules and processes, each constructed independently of the others by one or more individuals or groups. 6b1

Many implementation details of a module or process are basically of concern only to the implementers, and are influenced by a variety of factors, including the characteristics of the host on which it must run and the programming styles of its programmers. 6b1a

However, three characteristics of each module or process are of wider concern and must be rigidly specified and that specification adhered to: 6b1b

1) The module's or process' function. 6b1b1

2) The format of any data bases it requires as input or generates as output. 6b1b2

3) The manner in which the module or process is

accessed by, and communicates with, other modules or processes. 6b1b3

The conventions that specify the above for a specific function are called a protocol. 6b1c

In the course of building the NSW, protocols must be specified that govern a number of areas (the need for additional protocols will no doubt be exposed during the course of the contract year). 6b2

The design of protocols for the NSW will be a shared responsibility with other NSW contractors roughly as described below. Management of the NSW protocol developments is an ARC responsibility and is described later. 6b3

Areas of ARC Prime Design Responsibility 6b4

1) The NSW is dependent upon the following existing Network protocols, some of which may require modification or replacement for NSW use: 6b4a

a) Host-Host Protocol 6b4a1

b) File Transfer Protocol 6b4a2

c) Telnet Protocol 6b4a3

d) Graphics Protocol 6b4a4

2) The format of the data base describing the Command Language Grammar for a tool. 6b4b

3) The format of the User-Profile data base. 6b4c

4) Handling displayed information on a multiwindow, two-dimensional display screen. 6b4d

The latter three areas are tasks associated with the development of the NSW frontend, but must be coordinated as protocol developments. 6b4e

Areas of Other Contractor Prime Design Responsibility, or Shared Responsibility With ARC 6b5

1) The use of tools (by the frontend, the Works Manager, and other tools). Final responsibility for tool

selection for the initial NSW system must rest with the NSW Steering Committee. Prime responsibility for design of protocols between tools and the Works Manager rests with Massachusetts Computer Associates (MCA). The responsibility for the protocols between the frontend and Works Manager, and frontend and tools is shared by ARC and MCA. The following tools have been identified: 6b5a

a) One or more Batch (RJE) Processors 6b5a1

b) A Text Editor 6b5a2

c) A Message Facility 6b5a3

d) The Programmer's Assistant 6b5a4

e) A Behavior Reporting System 6b5a5

f) A Project Management System 6b5a6

2) The interaction between a B3500 or 360 batch machine, its NSW minicomputer frontend, and the Works Manager. 6b5b

This task is related back to Host-Host protocol development (ARC responsibility) and to tool protocols (MCA and ARC responsibilities as above). 6b5b1

3) Interaction with the Datacomputer (MCA responsibility). 6b5c

Protocol Management Responsibilities 6b6

ARC will assume responsibility for the administration of NSW protocol development, and expects that responsibility to manifest itself in the following areas as funding permits: 6b6a

1) Management of the NSW protocol development process. In this role, ARC would serve as coordinator for NSW protocol development and, specifically: 6b6a1

a) Participate in the specification of (or, where necessary, specify) standards for protocols themselves and for their measurement and documentation. 6b6a1a

b) Participate in the initial identification of

protocols and the ensuing allocation of design responsibility. 6b6a1b

c) Assure that the design process is preceded by a written statement of the protocol's use and character. 6b6a1c

d) Follow through on whatever process is established to get protocols approved by the ARPANET community for general implementation. 6b6a1d

e) Coordinate the implementation of NSW protocols. 6b6a1e

f) Schedule phase-overs to revised versions of protocols. 6b6a1f

2) Evaluate protocols and their implementations. In this role, ARC would subject protocols and their implementations to a program of regular evaluation and testing to identify design flaws or areas for improvement, and to determine the correctness and effectiveness of specific implementations. Specifically, ARC would: 6b6a2

a) Compare related protocols for possible duplication of function and suggest improvements. 6b6a2a

b) Investigate the feasibility of conducting theoretical analysis of protocols. 6b6a2b

c) Measure the frequency with which various protocol features are used and thus identify the areas that would most benefit from increases in efficiency. 6b6a2c

d) Periodically test specific protocol implementations for correctness and completeness, and provide certification of an implementation's status. 6b6a2d

e) Measure and evaluate the performance of specific protocol implementations. 6b6a2e

3) Maintain documentation for NSW protocol and implementation. In this role, ARC would collect or, where necessary, generate documentation for protocols and their implementations, and make available the

information it obtains from its analysis of them, ARC would: 6b6a3

a) Publish protocol, documentation, and measurement standards. 6b6a3a

b) Tag published protocol documents with their official designation -- experimental, official, and so forth. 6b6a3b

c) Publish test and measurement results for specific implementations, noting their completeness, any idiosyncrasies or bugs they might possess, their performance, and their official designation. 6b6a3c

d) Maintain lists of protocol designers and implementers. 6b6a3d

e) Route externally contributed implementation critiques to the appropriate parties. 6b6a3e

f) Provide consulting services to new hosts. 6b6a3f

4) ARC would perform trial implementations and experiment with protocols as needed and resources permit. 6b6a4

NLS as an NSW Tool 6c

The modifications and enhancements described in this section are desirable for the NSW version of the NLS workshop. To maximize their usefulness, ARC will work closely with the NSW's potential users. Trial features can be added, tested in the field, and modified as experience is gained. One of the goals is to make NLS an example of a tool fully integrated into the NSW environment. Tasks are listed below in approximate order of priority as we understand the needs. 6c1

1) Complete the split of the NLS system into frontend (user interface control) and backend (execution) parts to increase efficiency and to take full advantage of the NSW environment. 6c1a

2) Complete the work necessary to bring up the new NLS that is now in use at ARC, as the standard system on

OFFICE-1. This new NLS system is to be the basis for further development and use by NSW users. 6c1b

3) Perform those modifications to the NLS file system that may be necessary to integrate it most efficiently into the larger NSW file system and conventions; deal with problems of file transfer and use imposed by network bandwidth (this may involve partial paging of files across the network); consider modifications in the Datalanguage specifications to create the necessary features to handle NLS structured files. 6c1c

4) Package and provide NLS user interfaces of special interest to initial NSW users. 6c1d

a) Provide special user interface features to aid COBOL programmers in the preparation, study, and documentation of COBOL source programs. These features would provide shorthand ways to create source statements and program structuring, aid program online viewing and studying, and provide some simple syntax checking. 6c1d1

b) Provide a special user interface oriented toward the clerical worker involved in simple word processing activities who has no need for the complete richness of the general NLS system. 6c1d2

5) Provide preprocessing facilities for making NLS-created COBOL code, written and structured for the convenience of the programmer, satisfy the syntactic requirements of the particular destination COBOL compiler accessed through the Remote Job Entry facilities of the NSW Works Manager. 6c1e

6) Extend the current HELP system database for use with the COBOL programmer's and secretaries' interfaces. 6c1f

7) Interface the powerful formatting capabilities of the NLS Output Processor to work with the Linotron photocomposition device at Wright Patterson Air Force Base and provide other needed extensions to NLS facilities for document publication. 6c1g

Discussion of Above Tasks 6d

1) NLS Frontend/Backend Split 6d1

One important task that will remain unfinished at the end of the current ARPA contract is the reorganization of NLS into a "frontend" system that can reside on a minicomputer and perform the highly interactive tasks associated with command setup and feedback, and a "backend" system that resides on a TENEX and performs command execution. This task is important because: 6d1a

a) It allows the NLS system to fit nicely into the NSW environment that is expected to utilize a frontend system as described earlier, 6d1a1

b) It is estimated that such an organization can reduce the cost of NLS operation by about a factor of two. This decrease in usage cost is particularly important for the NSW program that will utilize NLS widely, 6d1a2

The reorganization that would take place would be such as to fully utilize the NSW frontend capabilities proposed above, 6d1b

It is worth pointing out that once a split has been completed between frontend and backend functions two important directions of evolution become open, 6d1c

a) Move the most commonly used editing and viewing functions to be co-resident with the frontend functions on a local minicomputer. This direction would lead to additional substantial cost-saving in usage of NLS, 6d1c1

b) Move backend functions to other computer systems, such as Multics or IBM 370 equipment. Given that the highly interactive frontend user interface functions are running on a minicomputer, other machines than a PDP-10 may be more appropriate for the backend functions. It should be a relatively straightforward task to move the backend functions to other systems, 6d1c2

2) Make the new version of NLS operational on the OFFICE-1 computer system, 6d2

Another essential task that will be incomplete at the end of the current contract is to make the new NLS system, now operational at ARC on an experimental basis, the standard NLS system at the OFFICE-1 NLS Utility. Involved here is: 6d2a

a) Completion of the evaluation and modification of new user features and command syntax. 6d2a1

b) Completion of the HELP database of online documentation. 6d2a2

c) Completion of debugging and quality assurance testing. 6d2a3

3) Integrate the NLS File System into the NSW File System 6d3

The NSW system-wide file system will be the responsibility of the NSW Works Manager. The Datacomputer is expected to play a central role in this system. 6d3a

The Datacomputer is planned for NSW use in such tasks as: 6d3b

a) A tertiary store for code and documentation. 6d3b1

b) A tool for a general information retrieval system used in, for example, the maintenance of a Program Library. 6d3b2

c) An information storage and retrieval component of a management control system. 6d3b3

A number of issues need resolution with respect to whether or not to store all files between sessions on the Datacomputer, or to store working material on storage media, such as disks, close to tools, such as NLS, when the files are being primarily used with that tool. 6d3c

There are trade-offs in design complexity in NSW tools -- such as NLS and the NSW Works Manager vs required ARPANET and Datacomputer bandwidth -- that need to be considered. 6d3d

ARC would work with MCA (the contractor for the Works Manager, with prime responsibility for the NSW file system) to consider such trade-offs relative to use of NLS and determine what limitations in Datalanguage and Datacomputer implementations might impact on use of NLS in the NSW environment. ARC would make appropriate recommendations and make NLS file system modifications

as needed to integrate properly into the larger NSW file systems environment. 6d3e

4) Package NLS Functions and User Interface Features: Interfaces for COBOL Programmers, Programmer Support Personnel, and Others 6d4

NLS, and in particular its display version, has been used primarily by its developers. These users are highly skilled and motivated. They use the system in varying amounts, up to eight or more hours per day, depending on their tasks. These users were screened for employment because of their interest in applying computer technology and their expected ability to feel comfortable, and contribute, in a highly computerized environment. 6d4a

The user population outside of ARC contains a wider mix of people with different tasks, motivations, skills, and attitudes toward interactive computer use. These users need to have the NLS command set and training packaged in various ways, to simplify learning, and to be more explicitly aimed at their tasks where appropriate. 6d4b

The user interface is currently implemented with a Command Meta-Language System (CMLS) that makes it easy to modify the interface, repackage the features for special functions, and tailor the interface to individual needs. The split of the NLS system into front (user interface) and back (execution) parts, possibly running on separate machines, increases the efficiency and usefulness of this design. A discussion of this split appears above. 6d4c

To meet the needs of individual or special groups of users we must clearly analyze and understand what problems they are having with the present versions of NLS and what features or functions would solve these problems. This will require close interaction with selected users, production of modified interfaces, and trial use by users with the modified interfaces. 6d4d

NLS is a very powerful and rich system. This richness is very important for experienced users, but can be confusing to those less experienced. Packaging NLS into self-contained subsets might better meet the needs of users of various experience levels. The syntax and command vocabulary can be

made to mesh more closely with the concepts and task vocabulary of selected user classes. 6d4d1

Two types of user interfaces of special interest to the NSW effort are those for COBOL programmers and for programmer support personnel. 6d4e

a) COBOL Interface 6d4e1

Characteristic problems encountered by COBOL programmers are: 6d4e1a

1) The misspelling of lengthy reserved words, paragraph and section names, and data identifiers 6d4e1a1

2) Insufficient comments for data definitions sections, and paragraphs 6d4e1a2

3) Inefficient data definitions 6d4e1a3

4) Excessive verbosity 6d4e1a4

5) Lack of program structuring conventions and enforcement for increased comprehension 6d4e1a5

6) The time consumed in submitting a job for compilation and execution, only to find some trivial syntax error or some undefined data element. 6d4e1a6

ARC proposes that it develop a COBOL programmer's interface to NLS that will allow the user to enter and edit COBOL source code with the following features: 6d4e1b

1) Under user control, automatically complete the spelling of COBOL reserved words while inserting COBOL source code 6d4e1b1

2) Automatically apply structure to the inserted source code 6d4e1b2

3) Help ensure syntactic correctness of the inserted source code 6d4e1b3

4) Attempt to use existing data definitions to 6d4e1b4

a) Provide spelling completion when requested 6d4e1b4a

b) Request new definitions and documentation when a new identifier is used by the programmer. 6d4e1b4b

Such an interface could be easily implemented in the Command Meta-Language version of NLS as a special grammar with execution functions that deal with the COBOL programmer's problems while imposing documentation standards. Details of such a COBOL programmer's interface will be coordinated closely with personnel at the Air Force Data Systems Design Center. Similar interfaces may be developed for other higher level languages. 6d4e1b5

b) Programmer Support Personnel Interface 6d4e2

A special simplified interface to NLS would be provided to handle the needs of clerical support staff. It could be available in an offline version, as well as an online version, to make most effective use of computer time. It would be designed to be usable by temporary clerical help with a minimum of training; it would be as self-teaching as possible. 6d4e2a

5) Preprocessing of Source Code Prepared for Remote Job Entry for COBOL Compilation 6d5

A Remote Job Entry (RJE) facility will be made available through the NSW Works Manager. This facility will allow the NSW user to, among other things, compile and execute his COBOL programs on various machines available within the NSW environment that provide the required RJE facilities. An NLS preprocessor would take the source code prepared by the NLS COBOL programmmer's interface and format it appropriately for further handling by the NSW Works Manager. 6d5a

6) HELP-Tutorial Databases and Other Documentation 6d6

NLS user documentation will exist in two forms: online and in appropriate hardcopy forms. 6d6a

Hardcopy documentation will exist in the forms agreed

between ARC and ARPA, such as cue cards, introductory guides, scenarios, or more extensive user guides. 6d6b

Online documentation and HELP databases will exist with appropriate easy-to-learn methods of access and browsing. Some simple tutorial facilities may be available as required. 6d6c

More sophisticated NLS tutorial and HELP facilities can evolve in later NSW system versions. 6d6d

7) NSW Publication System 6d7

One of the important early uses of NLS will be to produce publication quality documents in hardcopy or microfiche. Features required for these functions will be packaged as a separately identifiable NSW tool. 6d7a

Enhancements to NLS to produce documents in selected DoD formats will be made as funding permits. 6d7b

The Output Processor, a powerful hardcopy text formatter used in conjunction with NLS, will be modified so that it can be used with the DoD's Linotron phototypesetter. This will allow NSW users to write documents using NLS and then format them for publication using the Linotron. The Output Processor now supports hardcopy terminals, line printers, and Comp-80 COM devices, which produce publication quality hardcopy as well as microfiche and microfilm. 6d7c

NSW Host Computer Connection (An Optional Task) 6e

The NSW Host Connection to the ARPANET will be a combination of hardware and software designed for communication with a specific host. The hardware will include a minicomputer that will perform the basic tasks needed for communication: generation and interpretation of the proper handshaking codes and ARPANET protocols, receipt of data and command streams from remote job entry stations and online terminals, and communication with the NSW Works Manager and other facilities. 6e1

In particular, the Host Connection must receive and process commands sent to it by the NSW Works Manager and subsidiary modules responsible for defining and validating requests for tool use and seeing that the requests are carried out. 6e2

Design Considerations 6e3

Hardware 6e3a

Connection software should run on the standard minicomputer chosen for the NSW frontend, probably a PDP-11, connected to the network through an IMP via an IMP-Host interface. 6e3a1

The minicomputer would connect, where possible, directly to a host's (high speed) data channel. 6e3a2

Where lease agreements or hardware incompatibility restrict direct connection, a special hardware interface to the channel could be supplied by the manufacturer of the host computer. This hardware could make the minicomputer look like a device known to the host. 6e3a3

An evaluation should be made as early as possible as to whether or not the Host Connection computer and frontend computer should be the same or separate machines. 6e3a4

Software 6e3b

The design should avoid modifications to the host computer's operating system, to as great an extent as is possible. Maintenance of such modifications is an endless and unrewarding chore and generally delays

installation of new and desirable releases of the operating system. 6e3b1

In addition it would make the task of interfacing to hosts with different manufacturers' hardware much more complex, and require a learning period for the implementors for each new operating system encountered as new hosts are added to the NSW. 6e3b2

An alternative is to make the minicomputer look like a device the host computer system already is programmed to handle. 6e3b3

The virtual device would not necessarily have to be the same for all hosts, although the connection software would probably be simplified if it were. 6e3b4

The device could appear to be a disk drive, tape drive, terminal, etc. Probably the easiest to generalize for a variety of hosts would be a terminal. However, an alternative should be made available to support hosts that offer only batch services and do not support terminals. 6e3b5

The data coming into the mini from the network, remote job entry stations, or terminals for processing and transmission to the host would include: 6e3b6

1) Files from other network hosts to be sent to the local host for storage, processing, local printing, etc. 6e3b6a

2) All types of text files. 6e3b6b

3) Pre-generated job control streams with pre-compiled object code to be run on the host. These data could come over the network from another host (e.g., Datacomputer), from a remote job entry station, or from the NSW Works Manager. 6e3b6c

4) Source code for a variety of compilers and assemblers that include job control data with it. 6e3b6d

5) Job control streams generated by the NSW supervisor. 6e3b6e

6) Job control and data streams from remote job entry stations, 6e3b6f

7) Messages and data coming from terminals, both through the network and from local connections, 6e3b6g

8) Messages, control data, and synchronization data coming from other hosts. An example is the Datalanguage interface, 6e3b6h

Data coming into the mini from the local host could include: 6e3b7

1) Files (source code, text, object code) to be transfered to another host for editing, compiling, or whatever, 6e3b7a

2) Messages, files, etc, to be sent to terminals, line printers, and any other device connected to the mini, 6e3b7b

3) Messages and data from terminals connected to the host, 6e3b7c

The software to process the data streams would operate under the chosen NSW PDP-11 operating system. The operating system should be the same as the one used for the frontend. In fact, it is possible that the frontend and the connection systems will run on the same machine, 6e3b8

The software should maintain all the status information necessary to reconnect to the host, the network, or the remote job entry station in event of the failure of any one of them, 6e3b9

Implementation 6e4

The system would have two major components. The first would be general for all hosts and include communication with the operating system and with the NSW, 6e4a

Additional modules in the general component would include processing of the General Network Protocols: File Transfer Protocol, Remote Job Entry Server, Telnet, and any special protocols that may developed, 6e4b

The second component could be a table-driven code that

depends on a data structure containing all necessary information about the local host. 6e4c

Initial implementation efforts should be directed towards getting the Burroughs 3500 at the Air Force Data Systems Design Center in Montgomery, Alabama, connected first to the Network and then to the NSW. 6e4d

The system software should be maintainable and debuggable to as large an extent as possible from a remote timesharing system. 6e4e

Proposal for Research No. ISU 74-132
Development Support NSW

WORK STATEMENT: 7

NSW Frontend development (Version 1 to be operational by July 1975) as funding and best efforts permit, 7a

1) Evaluate and select a minicomputer operating system, 7a1

2) Design and implement a frontend system to operate on PDP-11 and PDP-10 computers that will have the following features: 7a2

a) Operating system interfaces, 7a2a

b) Display and terminal control, 7a2b

c) Appropriate ARPANET Protocols, 7a2c

d) Command Language Interpreter driven by Command Language Grammar and User-Profile databases, 7a2d

3) Design a formal language for specifying NSW tool user interfaces and implement a compiler for it to produce the grammar databases mentioned above, 7a3

4) Provide user documentation in the form of an online HELP database and Command language reference manual summary, 7a4

5) Provide system documentation in the form of a system design overview and well commented source code listings, 7a5

6) Integrate the frontend system into the larger NSW system, 7a6

NSW Protocol Development and Management 7b

The following task areas will receive effort according to priorities agreed on with ARPA as funding permits, 7b1

1) Specify and publish standards for protocols, protocol measurement, and protocol documentation, 7b2

2) Maintain and publish, for each protocol, a statement of its intended use and character, the names of its designers, and its current state, 7b3

3) Maintain and publish design documents, tagged with their current state, 7b4

4) Evaluate existing protocols for possible duplication of function and perform re-design, as needed. 7b5

5) Follow through according to ARPANET procedures to get designated protocols approved as official for general implementation. 7b6

6) Schedule protocol implementations and phase-overs. 7b7

7) Evaluate protocol implementations; publish completeness, correctness, and performance data; investigate the feasibility of theoretical analysis; and serve as a clearinghouse for externally contributed evaluations. 7b8

8) Develop techniques for monitoring protocol usage. 7b9

9) Provide consulting services to new hosts. 7b10

10) Perform test implementations and trials as needed. 7b11

NLS as an NSW Tool 7c

1) perform the necessary changes to NLS that will enable it to operate as a cooperating, bona fide NSW tool, with a grammar for the frontend system and an execution module running under TENEX. 7c1

2) Make the new NLS system now running at ARC the standard NLS system at OFFICE-1. 7c2

3) Specify, design, implement, and document a COBOL programmer's text editing interface and a secretary's interface. 7c3

4) Create an NSW publication tool by appropriately packaging and enhancing existing NLS capabilities. Interface NLS to the Linotron phototypesetter. 7c4

5) Study and integrate the NLS file system into the larger NSW file system, as needed. 7c5

6) Provide a preprocessor for source code to be submitted to the Remote Job Entry facility of the NSW Works Manager for compilation and execution of COBOL programs. 7c6

7) Provide appropriate online and hardcopy user documentation. 7c7

NSW Host Computer Connection (An Optional Task) 7d

1) Design, implement and document the NSW Host Connection system and test thoroughly for smooth interface to the NSW and to the local host. Provide documentation at the system level in the form of a system overview and well-commented source code. 7d1

2) After study of host hardware and software, specify any additional hardware requirements and modification to Host's operating system. 7d2

3) Supervise the installation of the software at the Host site and testing of all phases of the system. 7d3

4) Determine whether the Host Connection hardware can and should be shared with that of the NSW frontend system. 7d4

5) Provide user documentation for local operating personnel as needed. 7d5

MILESTONES: 8

The project milestones below represent an initial estimate. More detailed milestones will be prepared for internal project control and will be available to ARPA if desired after the start of the contract. Dates are July 1,1974 to June 30,1975. 8a

NSW Frontend Development 8b

July 1 8b1

Task begins. 8b1a

August 1 8b2

Frontend PDP-11 configuration recommendations. Operating system choice made. Implementation language choice made. 8b2a

October 1 8b3

Command Meta-Language design complete. NSW display recommendations. Cross-net debugging and other tools needed for task completed. 8b3a

November 1 8b4

Terminal control module designed. 8b4a

December 1 8b5

Operating system interface completed. CML compiler completed. 8b5a

January 15 8b6

Terminal control module implemented on PDP-11 and PDP-10. Documentation begins. 8b6a

February 1 8b7

Command Language Interpreter, Version 1, running. 8b7a

April 1 8b8

First version of complete frontend system running. 8b8a

July 1 8b9

Version 1 for NSW release running and documented, 8b9a

NSW Protocol Development 8c

Milestones are difficult to specify for this task, because of its close interaction with other NSW contractors and ARPANET resources and somewhat open-ended consulting and evaluation nature, 8c1

July 1 8c2

Task begins, 8c2a

September 1 8c3

Last cut at specifying NSW protocols needed and issues that may require alteration of existing protocols, 8c3a

October 1 8c4

Draft designs for new protocols specified, 8c4a

November 1 8c5

Draft designs reviewed. Modifications to existing protocols recommended, 8c5a

NLS as an NSW Tool 8d

July 1 8d1

Task begins, 8d1a

September 1 8d2

Design for NLS frontend-backend split completed, 8d2a

October 1 8d3

NLS two fork split completed. New NLS released to OFFICE-1 and documented, 8d3a

November 1 8d4

NSW management begins discussion with ARC Applications Department to provide NLS service via an NSW frontend to initial NSW users, 8d4a

December 1 8d5

COBOL programmers interface designed, NLS file system changes to operate in full NSW environment determined, 8d5a

December 15 8d6

Secretarial interface specified, 8d6a

February 1 8d7

NLS grammar modified for NSW environment, version 1 COBOL programmers interface ready for initial trial use, Version 1 Secretary's interface ready for trial use, 8d7a

March 1 8d8

Linotron interface complete, COBOL and Secretary's interfaces initial trials complete, File system changes for NSW environment complete, 8d8a

May 1 8d9

Second stage COBOL and Secretary's interfaces complete, 8d9a

July 1 8d10

NLS integrated as running tool in NSW environment and documented, 8d10a

NSW Host Computer Connection (Optional Task) 8e

It is assumed that the FTP, NCP, and TELNET protocols are obtained with the operating system. 8e1

July 1 8e2

Task begins. 8e2a

August 1 8e3

Complete initial stage of problem study. 8e3a

September 1 8e4

Recommendation on whether or not frontend and Host connection machines should be the same machine or not. Design for B-3500 interface complete. 8e4a

November 1 8e5

Implementation of B-3500 interface complete. 8e5a

January 15 8e6

RJE server protocol design complete. 8e6a

March 1 8e7

RJE server protocol implementation complete. 8e7a

April 1 8e8

Works Manager interface design complete. 8e8a

May 15 8e9

Works Manager implementation complete. 8e9a

July 1 8e10

System integrated into NSW, and documented. 8e10a

PERSONNEL: 9

Included below are biographies of personnel likely to be involved with the tasks proposed. In addition, ARC will be hiring two to three additional software personnel, and an additional technical writer. Other ARC staff will be called on as needed. 9a

REFERENCES: 10

(1) Robert Balzer (USC-ISI), T. E. Cheatham (HARV-10), Stephen Crocker (ARPA-IPT), Stephen Warshall (MCA), Design of a National Software Works, USC/Information Sciences Institute, Marina del Rey, California, ISI-RR-73-16, November 1973, 25p, NIC 19208. 10a

(2) Stephen Warshall (MCA), NSW Framework, MCA Proposal No. P-7-D01; Revised February 26, 1974. 10b

(3) Robert Balzer (USC-ISI), NSW Frontend, Working Paper, May 1974. 10c

(4) Jack Goldberg (Editor), Proceedings of the Tri-Service Symposium on the High Cost of Software, Held in Monterey, California, Stanford Research Institute, September 17-19, 1973. 10d

(6) Douglas C. Engelbart, William K. English, "A Research Center for Augmenting Human Intellect", In: AFIPS Proceedings, Fall Joint Computer Conference, Vol. 33, 1968, p.395-410. NIC 3954. 10e

(7) D. C. Engelbart, Intellectual Implications of Multi-Access Computer Networks, Paper presented at Interdisciplinary Conference on Multi-Access Computer Networks, Austin, Texas, April 1970, 12p. NIC 5255. 10f

(8) Douglas C. Engelbart, Coordinated Information Services for a Discipline- or Mission-Oriented Community, Paper presented at Second Annual Computer Conference in San Jose, California, January 24, 1973, 12 December 1972, 7p, NIC 12445. 10g

(9) Douglas C. Engelbart, Richard W. Watson, James C. Norton, The Augmented Knowledge Workshop, Paper presented at the National Computer Conference, New York City, June 1973, 1 March 1973, 19p. NIC 14724. 10h

(10) Douglas C. Engelbart, "Design Consideration for Knowledge Workshop Terminals", In: Proceedings National Computer Conference, 1973, p.221-227. NIC 14851. 10i

(11) Richard W. Watson, Basic Issues and Beliefs for the Design of Knowledge Workshop Systems and Status of NLS with Respect to Them, 20 November 1973, 14p. J20367. 10j

(12) Richard W. Watson, Some Thoughts on System Design to Facilitate Resource Sharing, 20 November 1973, 5p, J20391. 10k

(13) Donald I. Andrews, "Line Processor:-A Device for Amplification of Display Terminal Capabilities for Text Manipulation", In: Proceedings of the National Computer Conference, 1974, p.479-487, NIC 20184, 10l

(14) Charles H. Irby, "Display Techniques for Interactive Text Manipulation", In: Proceedings of National Computer Conference, 1974, p.911-919, NIC 20183, 10m

(15) D. C. Engelbart and Staff of ARC (SRI-ARC), Computer-Augmented Management-System Research and Development of Augmentation Facility - Final Report, RADC-TR-70-82, April 1970, 268p, NIC 5139, 10n

(16) Augmentation Research Center, Stanford Research Institute, Online Team Environment: Network Information Center and Computer Augmented Team Interaction - Final Report, RADC-TR-72-232, 8 June 1972, 268p, NIC 13041, 10o

Proposal for Research No. ISU 74-132
Development Support NSW

11

I. ESTIMATED TIME AND CHARGES 11a

It is proposed that the work outlined herein be performed during a period of 12 months commencing 1 July 1974. 11a1

Pursuant to the provisions of ASPR 16-206.2, attached are a cost estimate and support schedules in lieu of the DD Form 633-4. Also enclosed is a signed form complete except as to the "Detailed Discussion of Cost Elements." 11a2

II. REPORTS 11b

The work proposed herein will result in three types of documentation. 11b1

1) Online user documentation for the systems and features proposed. 11b1a

2) Hardcopy user documentation, such as cue cards, to be agreed on with ARPA. 11b1b

3) A final report of the experience gained as appropriate. 11b1c

III. GOVERNMENT-FURNISHED EQUIPMENT 11c

The work proposed herein will require Government-Furnished Equipment (GFE). The equipment required is PDP-10 TENEX computer time on the ARPA-owned, but ARC-operated, computer facility between July 1, 1974, and December 31, 1974. Resources are required to support at least 8 ARC display terminals to be used simultaneously. ARC users should be able to get at least 35%-40% of the CPU cycles available to users during the normal working day (8:00 A.M. - 5:00 P.M., PDT) and access to the system in off hours as available. 11c1

After December 31, 1974, ARC will buy the PDP-10 TENEX time needed for its NSW tasks from the OFFICE-1 facility, since the machine operated at SRI by ARC will be moved by ARPA to another location. 11c2

IV. CONTRACT FORM 11d

Because of the nature of the work proposed, it is requested that any contract resulting from this proposal be awarded

on a cost-plus-fixed-fee basis, as a modification to the NSW TENEX Facility contract between SRI and RADC/ARPA resulting from SRI Proposal No. ISU 74-127. 11d1

V. ACCEPTANCE PERIOD 11e

This proposal will remain in effect until 1 August 1974. If consideration of the proposal requires a longer period, the Institute will be glad to consider a request for an extension of time. 11e1

Proposal for Research No, ISU 74-132
Development Support NSW

COST ESTIMATE FOR CORE PROPOSAL
FOR 12 MONTHS, STARTING 1 JULY 1974
1) NSW Frontend Development
2) NSW Protocol Development and Management
3) NLS as an NSW Tool

Personnel Costs

Supervision		
Prof		
Clerical		
Total Direct Labor		$ 181,892
Payroll Burden @ 28%		50,930
Total Labor and Burden		232,822
Overhead @ 107%		249,120
Total Personnel Costs		$ 481,942

Direct Costs *			
Travel			$ 7,797
4 trips, Montgomery, @ $329	=	$ 1,316	
11 trips, Washington, D.C., @ $355	=	3,905	
3 trips, Boston, @ $389	=	1,167	
8 days subsistence in Montgomery @ $30	=	240	
22 days subsistence in Washington, D.C, @ $42,50	=	935	
3 days subsistence in Boston @ $38	=	114	
Auto Rental 8 days @ $15	=	120	
Computer Facility and Terminals *			$ 158,061
Materials and supplies			450
Communication			700
Consultation			2,100
Documentation			3,293
Total Direct Costs			$ 172,401

Total Estimated Cost	$ 654,343
Fixed Fee	52,348
Total Estimated Cost plus Fixed Fee	$ 706,691

* See Schedules that follow.

Proposal for Research No. ISU 74-132
Development Support NSW

COST ESTIMATE FOR THE OPTIONAL HOST CONNECTION TASK
FOR 12 MONTHS, STARTING 1 JULY 1974

Personnel Costs

Supervision		
Prof		
Clerical		
Total Direct Labor		$ 17,368
Payroll Burden @ 28%		4,863
Total Labor and Burden		22,231
Overhead @ 107%		23,787
Total Personnel Costs		$ 46,018

Direct Costs *

Travel		$ 2,666
4 trips, Montgomery, Ala.		
@ $329	= $1,316	
30 Days Subsist.@ $30**	= 900	
Auto Rental 30 days @ $15	= 450	
Computer Facility and Terminals *		$ 25,731
Communication		200
Documentation		799
Materials and supplies		50
Total Direct Costs		$ 29,446

Total Estimated Cost	$ 75,464
Fixed Fee	6,037
Total Estimated Cost plus Fixed Fee	$ 81,501

* See Schedules that follow.
** Computer system installation expected.

Proposal for Research NO. ISU 74-132
Development Support NSW

COST ESTIMATE FOR ALL PROPOSED TASKS
FOR 12 MONTHS, STARTING 1 JULY 1974

Personnel Costs			
Supervision			
Prof			
Clerical			
Total Direct Labor			$ 199,260
Payroll Burden @ 28%			55,793
Total Labor and Burden			255,053
Overhead @ 107%			272,907
Total Personnel Costs			$ 527,960
Direct Costs *			
Travel			$ 10,463
8 trips, Montgomery, @ $329	=	$ 2,632	
11 trips, Washington, D.C., @ $355	=	3,905	
2 trips, Boston, @ $389	=	1,167	
38 days subsistence in Montgomery @ $30	=	1,140	
22 days subsistence in Washington, D.C. @ $42.50	=	935	
3 days subsistence in Boston @ $38.00	=	114	
Auto Rental 38 days @ $15	=	570	
Computer Facility and Terminals *			$ 183,792
Communication			900
Consultation			2,100
Documentation			4,092
Material and Supplies			500
Total Direct Costs			$ 201,847
Total Estimated Cost			$ 729,807
Fixed Fee			58,385
Total Estimated Cost Plus Fixed Fee			$ 788,192

* See Schedules that follow.

SCHEDULE A
DIRECT LABOR

Direct labor charges are based on the actual salaries for the staff members contemplated for the project work plus a 4% judgmental factor of base salary for merit increases during the contract period of performance. The precise factor applied is dependent on the estimated period of performance. Frequency of salary reviews and level of merit increases are in accordance with the Institute's Salary and Wage Payment Policy as published in Topic No. 505 of the SRI Administration Manual and as approved by the Defense Contract Administration Services Region.

SCHEDULE B
OVERHEAD AND PAYROLL BURDEN

These rates have been found acceptable by the Department of Defense for billing and bidding purposes for the calendar year of 1974. We request that these rates not be specifically included in the contract, but rather that the contract provide for reimbursement at billing rates acceptable to the Contracting Officer, subject to retroactive adjustment to fixed rates negotiated on the basis of historical cost data. Included in payroll burden are such costs as vacation, holiday and sick leave pay, social security taxes, and contributions to employee benefit plans.

SCHEDULE C
TRAVEL COSTS, COMMUNICATION, MATERIAL AND SUPPLIES

Travel

Air fare is based on prices for travel to Washington D.C. at $355, to Montgomery, Alabama, at $329, and to Boston at $389, round trip tourist, established in the Official Airline Guide dated April 1, 1974.

Domestic subsistence rates and travel by private auto are established standards based on cost data submitted to and approved by DCAA.

Communication

This is an engineering estimate of the toll charges for

telephone calls during the period or performance.

Materials and Supplies

This is an engineering estimate of the cost of materials and supplies based on previous experience with similar projects.

SCHEDULE D
CONSULTANTS

Although the individuals have not yet been selected, it is anticipated that consultation services will be required in this project. The estimated amount is based on seven days of consulting at a rate of $300 per day.

SCHEDULE E
DOCUMENTATION COSTS

Report costs are estimated on the basis of the number of pages of text and illustrations and the number of copies of reports to be produced, in accordance with the following rates per page which have been reviewed by DCAA:

Editing	$2.29
Composition	2.22
Coordination	.63
proofreading	.92
Illustration	19.14
Press and Binding	.021 per impression

The following is a breakdown of the estimated cost of report production:

Printing, 473 pages at $ 6.06 per page = (including editing, composition, report coordination, proofreading)	$ 2,865
Illustrations, 40 @ $19.14 =	767
Press and bindery at, 21,900 printed pages @ $ .021 per printed page =	460
Total Estimated Documentation Costs	$ 4,092

SCHEDULE F
COMPUTER SUPPORT COSTS

1) PDP-10 TENEX Computer Time	
a) July 1, 1974 to December 31, 1974	GFE
b) January 1, 1975, to June 30, 1975	
5.5 job slots for 6 months *	
16,765 x 5.5 =	$ 92,208
2) PDP-11 Systems	
a) Equipment (lease)	
1) PDP-11 Development Machine	
$1,531/mo x 12 =	18,372
2) Parity Memory add-on costs	
$500 + ($438.06/mo x 6) =	3,128
3) Network Interface =	11,050
4) PDP-11 Terminal access to ARPANET,	
$1278/mo x 6 =	7,668
5) Special cables, hardware, etc,	
$2,000 (estimated) =	2,000
b) Maintenance (16 hrs/d, 5 d/wk)	
1) DEC PDP-11,	
$500/mo x 18 =	9,000
3) Terminals	
a) NLS workstations (7)	
1) Display (7)	
$233/mo x 7 x 10 =	16,310
b) TNLS Terminals	
1) TI (incl maintenance) (4)	
$165/mo x 4 x 12 =	7,920
2) Maintenance, owned TI's (8)	
$20/mo x 8 x 12 =	1,920
3) Acoustic couplers (8)	
$16/mo x 8 x 12 =	1,536

c) Modems

1) Dial-up (7)
$36/mo x 7 x 12 = 3,024

d) Leased lines

1) DIA, data
$366/mo x 12 = 4,392
2) DIA, voice
$22/mo x 12 = 264

4) Tasker Display System (10 units for 3 mo)

a) Parts (estimated) = 3,000

5) Miscellaneous (estimated) = 2,000

Subtotal (Items 2-5) $ 91,584

Total (Items 1-5) $183,792

* UTILITY COMPUTER SUPPORT SUBCONTRACT COSTS:

The estimate given here is based on current costs as shown in SRI Proposal No. ISU 74-69, dated 26 April 1974. It is expected that costs as of January 1, 1975, will be about 20% lower. Cost per slot for 6 months estimated to be $16,765.

The estimate of the need for 5.5 Job Slots is based on experience at ARC on the number of console hours used by programming, documentation, and management people.

Proposal for Research No. ISU 74-132
Development Support NSW

1 JULY 1974
SRI-ARC 23352

Proposal For Research
SRI No. ISU 74-132

NATIONAL SOFTWARE WORKS DEVELOPMENTS

part One---Technical proposal

Prepared for:

Information Processing Techniques Office
Advanced Research Projects Agency
1400 Wilson Boulevard
Arlington, Virginia

Attention: Stephen D. Crocker

Prepared by:

Richard W. Watson,
Assistant Director
Augmentation Research Center

Approved:

Douglas C. Engelbart, Director
Augmentation Research Center

Bonnar Cox, Executive Director
Information Science and Engineering Division
Stanford Research Institute

1 JULY 1974
SRI-ARC 23352

Proposal For Research
SRI No. ISU 74-132

NATIONAL SOFTWARE WORKS DEVELOPMENTS

Part Two - Contractual Provisions

Prepared for:

Information Processing Techniques Office
Advanced Research Projects Agency
1400 Wilson Boulevard
Arlington, Virginia,

Attention: Stephen D. Crocker

KIRK 12-SEP-74 21:55 23960

Insert Sendmail form doesn't work

(J23960) 12-SEP-74 21:55;;;; Title: Author(s): Kirk E. Kelley/KIRK;
Distribution: /BUGS([ACTION]) FDBK([INFO-ONLY]) ;
Sub-Collections: SRI-ARC BUGS; Clerk: KIRK;

KIRK 12-SEP-74 21:55 23960

Insert Sendmail form doesn't work

"NLS internal error, string too long" 1

KIRK 13-SEP-74 03:12 23961

Some items for the NLS-NSW shopping list.

(J23961) 13-SEP-74 03:12;;;; Title: Author(s): Kirk E. Kelley/KIRK;
Distribution: /NPG([ACTION]) DVN([ACTION]) SRL([INFO-ONLY])
; Sub-Collections: SRI-ARC NPG; Clerk: KIRK;

KIRK 13-SEP-74 03:12 23961

Some items for the NLS-NSW shopping list,

I might not be able to make the meeting at 10:30 today, I was working on the FDBK file until the time on this citation having discovered FDBK was not considered in the shopping list and it (FDBK) had not been updated in over a month, These suggestions, in the most part, come from syntheses of suggestions in the FDBK file, Please consider them,

KIRK 13-SEP-74 03:12 23961

Some items for the NLS-NSW shopping list.

Full NLS editor. Expanded features might also include: 1

Implement or interface an automatic spelling program. 1a

Make the site field work (windowing or copying in appropriate instances) for remote file access in links. 1b

"Included" links under viewspec control to allow cross-file substitutions, editing, and searches. 1c

Augmented Feedback, Analysis, and Decision Process: 1/8-1/4 person year. 2

An ongoing, effective, responsive, user feedback-decision process is vital to the survival of any living system. 2a

"Ongoing" means day-to-day, week-to-week, month-to-month, version-to-version of NLS. 2b

"Responsive" implies automatic user voting and weighted opinion gathering mechanisms as well as easy access to the current status of suggestions and projects. 2c

"Effective decision" implies analysis. 2d

This would also include date, person-hour, and cost projections for appropriate features and projects. 2e

Simple, easy too use, information retrieval system with a database consisting of an easily queryable integrated online handbook. 1/2-1 person year. 3

contents of the handbook could be: 3a

Work Manuals with basic conventions and Step-by-step algorithms for accomplishing and automating specific tasks. 3a1

Userguides for general purpose learning. 3a2

An open-ended glossary, thesarus of special terms. 3a3

A dictionary of standard definitions 3a4

An index of keywords 3a5

An interface to online help facilities. 3a6

Online document locator for important or special documents. 3a7

Some items for the NLS-NSW shopping list,

Decisions and Decision Algorithms for feedback, analysis, and decision operations. 3a8

Accounting and financial information. 3a9

Catalogs of available online and offline materials. (Books, reports, articles, video-tapes, etc. 3a10

Directories of individuals, groups, and organizations with published online materials by author. 3a11

DOug Englebart's thoughts on the NLS Handbook. References... 3a12

"Integrated" means file independant. Cross-file searching, linking. 3b

A back-link capablity. 3c

Make the site field work (windowing or copying in appropriate instances) for remote file access in links. 3d

Simple, easy to use, information retrieval system in this application means it must be general enough to operate on the average NLS file WITHOUT the user being aware of or needing to know about structural relationships, view specification codes and file size limitations. It would require the minimum number of button pushes to use for this application. 3e

MEH 13-SEP-74 06:46 23963

OFFICE-1, The Equipment We Initially Started With.

(J23963) 13-SEP-74 06:46;;;; Title: Author(s): Martin E. Hardy/MEH; Distribution: /JCN([INFO-ONLY]) ; Sub-Collections: SRI-ARC; Clerk: MEH; Origin: < HARDY, OFF1-SYS,NLS;6, >, 13-SEP-74 06:35 MEH
;;;;####;

MEH 13-SEP-74 06:46 23963

OFFICE-1, The Equipment We Initially Started With.

Jess Hill:

Here is the list of equipment we first started with at Office-1. It is not necessarily what we would consider the optimum or minimum system configuration for running NLS but is the one we first started with. If I can be of further assistance my mailbox is HARDY@SRI-ARC.
...martin.. 1

OFFICE-1 INITIAL HARDWARE INVENTORY 2

Digital Equipment Corp.: 2a

KA10 Arithmetic Processor
KM10 Fast Register
KT10A Dual Mem Protect Relocate
TM10A Mag Tape Control
DC10A Data Line Scanner Control
TU10F 7-Channel Mag Tape (two)
DC10B 8-Line Group Unit
MF10 64-K Core Memory (two)
MC10 Memory Ports (eight)
DF10 Data Channel (two)
RP10C Disk Controller
RC10 Drum Controller
RM10B Drum
LP10D Line Printer 2a1

Other equipment: 2b

BBN Pager
BBN ARPANET interface
CALCOMP 268 Disk (four) 2b1

RLL 13-SEP-74 11:07 23964

use same line for optional info in show directory command

(J23964) 13-SEP-74 11:07;;;; Title: Author(s): Robert N. Lieberman/RLL; Distribution: /FDBK([ACTION]) ; Sub-Collections: SRI-ARC; Clerk: RLL;

RLL 13-SEP-74 11:07 23964

use same line for optional info in show directory command

Please put size info when using show directory with options on sam line as name o file, otherwise it goes to o fast on the screen, same for more than one copy of same file, 1

31051 Distribution
Phil Feldman, Lawrence H. Day,

DVN 13-SEP-74 13:31 23965

DUMB!

(J23965) 13-SEP-74 13:31;;;; Title: Author(s): Dirk H. Van Nouhuys/DVN; Distribution: /DIRT([INFO-ONLY]) ; Sub-Collections: SRI-ARC DIRT; Clerk: DVN;

DVN 13-SEP-74 13:31 23965

DUMB!

I feel I must imortalize this interaction between a real user and NLS. The user is 15, named LiseMcGilvery, very bright, and not at all knowledgeable about computers, With some brief instruction from me and some help from Help I had asked her to try to send me a journal message, in which, by the way, she succeeded . I had asked her to make notes on the TI paper where she had problems, and the title of this Item is what she scrawld in large letters beside this part of the roll. She was logged in as me, knew my ident, and was using my TNLS which emplys a special charactere echoing with '! instead of <CR> for command accept.
---------- 1

SEND C: Interrogate OK: 2

(send for action to:) T:/[A:]: dvn! 3

(send for information only to:) T:/[A]: jml! 4

(title:) T:/[A]: computers! 5

(type of source:) C: Statement (at) A:/[T]: I am finally beginning yo
un
**derstand these things! 6

I? A: me! 7

ME? A: Lise! 8

LISE? A: myself! 9

MYSEF? A: dvn! 10

DVN? A: forget it! 11

FORGET? A: Do not remembe this! 12

DO? A: 13

x# 14

RLL 13-SEP-74 13:59 23966

Non delivery of author copy on 3 and 11 Sept 74

(J23966) 13-SEP-74 13:59;;;; Title: Author(s): Robert N. Lieberman/RLL; Sub-Collections: SRI-ARC; Clerk: RLL;

RLL 13-SEP-74 13:59 23966

Non delivery of author copy on 3 and 11 Sept 74

I have been using NLS8 at ARC (not WORK).

RLL 13-SEP-74 13:59 23966

Non delivery of author copy on 3 and 11 Sept 74

I have found that two recently sent (authored) journal items were not delivered to my author branch in my initial file. I have had no file problmes with the file and it is not a 'bad file', this has happened for items sent 3 sept and 11 sept. I checked ad they were indeed sent to those on the distribution list. Ther may have been others authoreed that were not delivered to me (author) but I don't recall them. ONe item sent today (13 Sept) did show up in author branch as normal.

1

DVN 13-SEP-74 14:19 23967

Nice Sentence

(J23967) 13-SEP-74 14:19;;;; Title: Author(s): Dirk H. Van Nouhuys/DVN; Distribution: /KJM([ACTION]) ; Sub-Collections: SRI-ARC; Clerk: DVN;

DVN 13-SEP-74 14:19 23967

Nice Sentence

The flexible NLS command language and the hierarchical file structure, which are the basis of NLS power in sophisticated appoication, are hard for temporary people or occasionaly users to learn too use comforably (although the joffline text entry method mentioned above, DEX, can be learned by the average typist in about half and hour.). 1

DVN 13-SEP-74 16:57 23968

Documentation status

(J23968) 13-SEP-74 16:57;;;; Title: Author(s): Dirk H. Van Nouhuys/DVN; Distribution: /KIRK([ACTION] would you ask Anne weinberg to puttogether that list of documents?...that with learning the system should occupy her) JOAN([ACTION]) JMB([ACTION]) JHB([INFO-ONLY]) JCN([INFO-ONLY]) RWW([INFO-ONLY]) NDM([INFO-ONLY]) ; Sub-Collections: DIRT SRI-ARC; Clerk: DVN;

Documentation Status

Documentation staus during next week 1

The cue card has gone back to the printer for corrections. A color proof should emerge. Kirk....would you proff it when Will Ashworth calls.....Joan, would you alert Kirk when Wil calls? 1a

Kirk, the (documenation, howto,) is ready to go to DDSI when you run the program on it. Please do so when you have the alphabetic part of the glossary ready. I have not written an intrroduction. Take a swing at it if you want. I have given a draft of the howto to the two Jims. 1b

I hope a final draft of the primer is ready in (vannouhuys,fixedprim,) . Dean will send it to DDSI when he is ready with some other stuff. 1c

Jeanne Beck is finishing off the conversion table from old to new comands, will notify the two Jims of drafts early next week. 1d

JMB 14-SEP-74 14:16 23969

Help needed in describing NLS-8's Identification subsystem

(J23969) 14-SEP-74 14:16;;;; Title: Author(s): Jeanne M. Beck/JMB;
Distribution: /CHI([ACTION]) MLK([ACTION]) ; Sub-Collections:
SRI-ARC; Clerk: JMB;

JMB 14-SEP-74 14:16 23969

Help needed in describing NLS-8's Identification subsystem

Charles or Marcia, if one of you knows about the above New NLS Ident stuff, please respond via Journal, or pass it to someone who does,
--Jeanne

JMB 14-SEP-74 14:16 23969

Help needed in describing NLS-8's Identification subsystem

I'm trying to write a document for RWW & Operations that gives the NLS-8 equivalents of all Old NLS (NLS-7) commands. I'm having some trouble translating the Identification commands because I don't know what some of the new ones do. Could you look over this list of the old commands and verify that those that follow the dashes are the new commands with the same function? What are the new command(s) for the ones marked with ??????????????? 1

---- NLS-7 ---------- NLS-8 ---------- 2

execute identification submode <TNLS> ---------- Goto Identification 3

enter identification ---------- [Identification] Add (new record) 3a

modify record for ---------- [Identification] Load (record for) 3b

access ---------- ??????????????? 3b1

affiliation ---------- [Identification] Organization 3b2

capabilities ---------- [Identification] Capabilities 3b3

coordinator ---------- [Identification] Coordinator 3b4

delivery ---------- [Identification] Delivery 3b5

expand normal references ---------- [Identification] Expand 3b6

function ---------- [Identification] Function 3b7

group membership ---------- [Identification] Groups 3b8

host ---------- [Identification] Mail 3b9

id ---------- ??????????????? 3b10

Help needed in describing NLS-8's Identification subsystem

imp ---------- ????????????? 3b11

mailing address ---------- [Identification] Mail 3b12

miscellaneous comments ---------- [Identification] Comments 3b13

name ---------- [Identification] Name 3b14

phone ---------- [Identification] Phone 3b15

secondary affiliation ---------- [Identification] Secondary 3b16

status ---------- [Identification] Status 3b17

subcollections ---------- [Identification] Subcollections 3b18

update ---------- ????????????? 3b19

user ---------- ????????????? 3b20

verify ---------- ????????????? 3b21

status for ---------- [Identification] Show Record (for ident) 3c

verify identfile ------- [Identification] Verify (Master Ident-File) 3d

KIRK 14-SEP-74 15:46 23970

Bug with sendmail creates a file for items under 2000 characters

(J23970) 14-SEP-74 15:46;;;; Title: Author(s): Kirk E. Kelley/KIRK;
Distribution: /BUGS([ACTION]) KJM([ACTION]) ; Sub-Collections:
SRI-ARC BUGS; Clerk: KIRK;

KIRK 14-SEP-74 15:46 23970

Bug with Sendmail creates a file for items under 2000 characters

A journal file is created for even the smallest of messages if it happens to come from a statement already in a file instead of being typed in. If this disc space waster and inconvenience for the user is a feature instead of a bug, or if it is not going to be fixed, please let me know so I can change the documentation. 1

KIRK 14-SEP-74 16:53 23971

Is acurate NLS documentation of the 'frozen' NLS important?

(J23971) 14-SEP-74 16:53;;;; Title: Author(s): Kirk E. Kelley/KIRK; Distribution: /NPS([ACTION]) DIRT([ACTION]) ; Sub-Collections: SRI-ARC DIRT; Clerk: KIRK;

KIRK 14-SEP-74 16:53 23971

Is acurate NLS documentation of the 'frozen' NLS important?

Does bootstrapping mean ' ... I'm all right, so it's OK,' ?

KIRK 14-SEP-74 16:53 23971

Is acurate NLS documentation of the 'frozen' NLS important?

The way changes are being made in the "frozen" NLS and "discovered" by the NLS documenters implies to me that having correct, up-to-date documentation for NLS is not very important. If all those responsible for making these changes in this way agree that acurate NLS documentation is not important, I will try to cease getting so upset when I accidentally discover fundamental changes. 1

If there is anyone making a change who does happen to want it documented (advertised), they should somehow make sure at least a notification (preferably a design specification) of the change gets into <NLS,MODS,> as soon as possible after discussion with those who are concerned with the change (the documenters maybe?); and that the change gets moved to the DONE branch of <NLS,MODS,> after it has been made. 2

KIRK 14-SEP-74 18:54 23972

Is acurate NLS documentation of the 'frozen' NLS important?

(J23972) 14-SEP-74 18:54;;;; Title: Author(s): Kirk E. Kelley/KIRK;
Distribution: /NPG([INFO-ONLY]) ; Sub-Collections: SRI-ARC NPG;
Clerk: KIRK;

KIRK 14-SEP-74 18:54 23972

Is acurate NLS documentation of the 'frozen' NLS important?

(this is a secondary distribution, I said NPS instead of NPG for Distribution.)

KIRK 14-SEP-74 18:54 23972

Is acurate NLS documentation of the 'frozen' NLS important?

The way changes are being made in the "frozen" NLS and "discovered" by the NLS documenters implies to me that having correct, up-to-date documentation for NLS is not very important. If all those responsible for making these changes in this way agree that acurate NLS documentation is not important, I will try to cease getting so upset when I accidentally discover fundamental changes. 1

If there is anyone making a change who does happen to want it documented (advertised), they should somehow make sure at least a notification (preferably a design specification) of the change gets into <NLS,MODS,> as soon as possible after discussion with those who are concerned with the change (the documenters maybe?); and that the change gets moved to the DONE branch of <NLS,MODS,> after it has been made. 2

KIRK 14-SEP-74 19:10 23973

Showing information that exceeds the view area

(J23973) 14-SEP-74 19:10;;;; Title: Author(s): Kirk E. Kelley/KIRK;
Distribution: /BUGS([ACTION]) ; Sub-Collections: SRI-ARC BUGS;
Clerk: KIRK;

KIRK 14-SEP-74 19:10 23973

Showing information that exceeds the view area

The last two times I've shown something, it has pushed the top line or so off the screen and then said Type CA for more CD to quit ... Typing CA shows nothing but the message "Type CA to continue". No more info. 1

KIRK 14-SEP-74 20:16 23974

NP for the Help database

(J23974) 14-SEP-74 20:16;;;; Title: Author(s): Kirk E. Kelley/KIRK;
Distribution: /NPS([INFO-ONLY]) DIRT([INFO-ONLY]) ;
Sub-Collections: SRI-ARC DIRT; Clerk: KIRK;

KIRK 14-SEP-74 20:16 23974

NP for the Help database

Since all commands are unique to three letters, Help could contain a list of statement names so that those letters are all the user has to type in order to see the command. This could be done for other command recognition modes as well except Terse Terse, (no way to tell if the user typed a space) Terse Anticipatory, and all others would work. In this way, you could type the same letters you use in your recognition mode and see the corresponding command. 1

KIRK 15-SEP-74 01:07 23975

Feedback needs updating

(J23975) 15-SEP-74 01:07;;;; Title: Author(s): Kirk E. Kelley/KIRK;
Distribution: /SRI-ARC([INFO-ONLY]) ; Sub-Collections: SRI-ARC;
Clerk: KIRK;

KIRK 15-SEP-74 01:07 23975

Feedback needs updating

Since no one has touched the user feedback in over a month, since we are in the process of making a "shopping list" of NLS needs and possibilities, and since we are trying to get all of the bugs out of NLS, I have taken the liberty of updating <FEEDBACK,FDBK,>. I will be thinking of ways this can be made a more ongoing system in hopes that the ARC decision process can be more responsive to the feedback from it's environment than it has in the past. A healthy feedback-decision process is an essential part of any living system. Please think of ways to use our "augmented" capabilities to augment the ARC decision process. It needs it. Read <FEEDBACK,FDBK,INTRODUCTION> for a basic plan that needs improvement. Always include FDBK in the distribution of suggestions and reports of bugs. FDBK is already a member of the following groups: NP (needs and possibilities): RWW CHI FDBK. BUGS: CHI HGL DSM FDBK. 1

RWW 16-SEP-74 09:52 23976

Collection of Dialog on a Maling list Superintendent

(J23976) 16-SEP-74 09:52;;;; Title: Author(s): Richard W. Watson/RWW; Distribution: /SDC2([INFO-ONLY]) CHI([INFO-ONLY]) JEW([INFO-ONLY]) JAKE([INFO-ONLY]) DCE([INFO-ONLY]) WEC([INFO-ONLY]) EKM([INFO-ONLY]) ; Sub-Collections: SRI-ARC; Clerk: RWW;

Collection of Dialog on a Maling list Superintendent

Collection of Dialog on a Maling list Superintendent 1

Below I collected the two messages dealing with a mail superintendent and my comment. We could provide such an interface between the NLS ident system and sndmsg but there are some potential problems tha we saw when we essentially had the ident system open to eveyone to modify. The fiel became full of garbage. Mispelled names, incorrect phone numbers and addresses etc so we had to close it and interpose a person to see tha the file had useful and meaningful info. We'll add such a task to our shopping list for NSw managemet but feel that any such informal capability is tricky and should be gone ahead with with care. Also although such an interface would be useful for keeping mailing lists and phone numbers to be useful with sendmsg one would need to be able to use the idents of somethiing else for the group name. 1a

13-SEP-74 0528-PDT CROCKER at USC-ISI: Automated mailing lists
Distribution: SUSSMAN AT BBN, WATSON AT SRI-ARC, crocker, carlson
Received at: 13-SEP-74 05:29:03 1b

Idea for a mailing list superintendent: 1b1

Have a program which reads messages directed to it from users who
wish to get on and/or off mailing lists. Have a separate program
which reads messages from users who wish to establish or disestablish
mailing lists. Messages sent to the first program need only the user's
mailing address (which is provided automatically) and the name of the
mailing list he wants to join. That name could
be the subject. The program can assume that if he's on he wants to
get off, and if he's off he wants to get on. The mailing list should
be accessible to anybody for read access. If read access is not
available, then the user will need mhanisms for testing
his membership and the body of theext may
be used. 1b2

The program should acknowledge its transactions by sending a message
back to the user. Another copy should be sent to the owner of

the list,
i.e. the guy who established the list. 1b3

If membership is not to be granted on demand of the user, the program
might forward a copy of the request to the owner for disposition
and so inform the user. This property should be set at establishment
time by the other program. 1b4

This pair of programs would provide most of the machinery required
to automate operation of the NWG, INWG, etc. 1b5

14-SEP-74 1136-PDT CROCKER at USC-ISI: Mailing list keeper upper
Distribution: WATSON AT SRI-ARC, crocker, sussman at bbn, carlson
Received at: 14-SEP-74 11:37:59 1c

Dick,
Thanks for your note. The most attractive features of sndmsg are: 1c1

1. It's installed on many machines right now; 1c2

2. No additional software is necessary to get at new facilities --
one just addresses messages to a new address; 1c3

3. The user is not tied up while activity takes place -- deferred
execution is the norm. 1c4

In a sense, what the NSW will provide is a single front-end which
provides access to an expandible set of functions. Removing the
need for centralized control is essential. Similarly, I think
it's important to have some mailing lists which do not require concurrence
from some authority in order to join. 1c5

Sooo... Is there any way to have your Ident system driven from sndmsg? 1c6

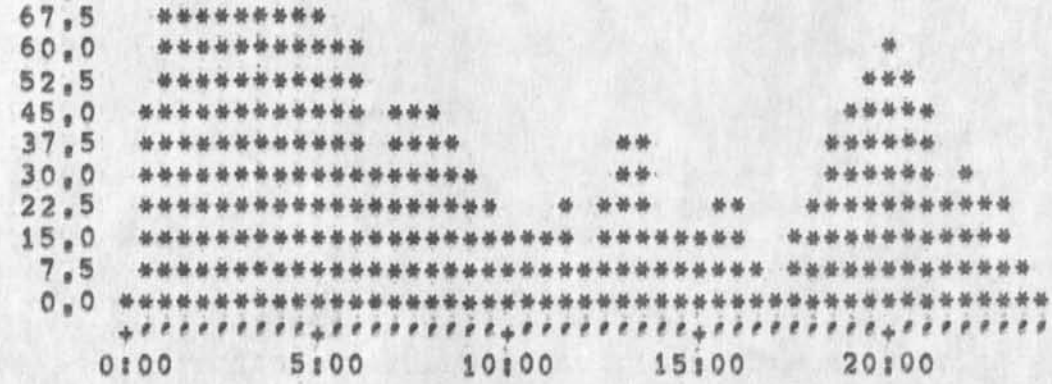
Steve 1c7

Superwatch Average Graphs for Week of 8/18/74

(J23977) 16-SEP-74 10:01;;;; Title: Author(s): Susan R. Lee/SRL;
Distribution: /JCN([INFO-ONLY]) RWW([INFO-ONLY]) DCE([
INFO-ONLY]) JCP([INFO-ONLY]) DVN([INFO-ONLY]) JAKE([
INFO-ONLY]) DLS([INFO-ONLY]) DSM([INFO-ONLY]) KJM([INFO-ONLY
]) ; Sub-Collections: SRI-ARC; Clerk: SRL; Origin: < LEE,
WEEK8/18GRAPHS.NLS;2, >, 16-SEP-74 09:59 SRL
;;;;<LEE>WEEK8/18GRAPHS.NLS;1, 4-SEP-74 11:49 SRL ;####;

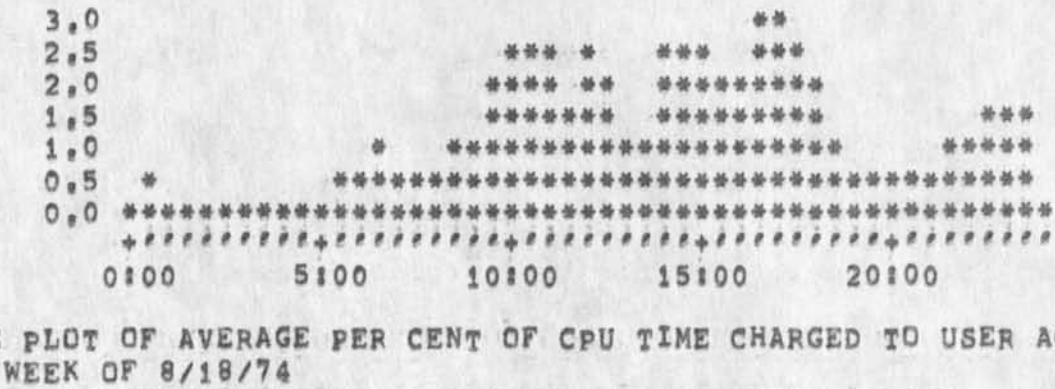
Superwatch Average Graphs for Week of 8/18/74

TIME PLOT OF AVERAGE IDLE TIME FOR WEEK OF 8/18/74
x axis labeled in units of hr:min, xunit = 30 minutes 1



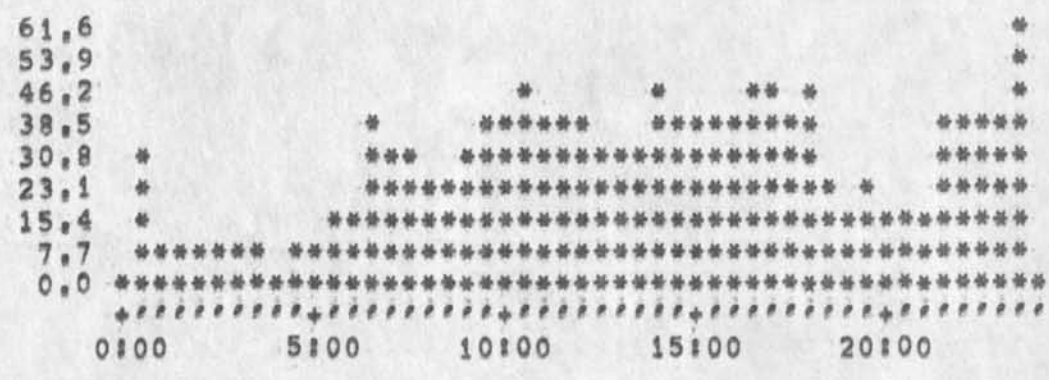
1a

TIME PLOT OF AVERAGE NUMBER OF GO JOBS FOR WEEK OF 8/18/74
x axis labeled in units of hr:min, xunit = 30 minutes 2



2a

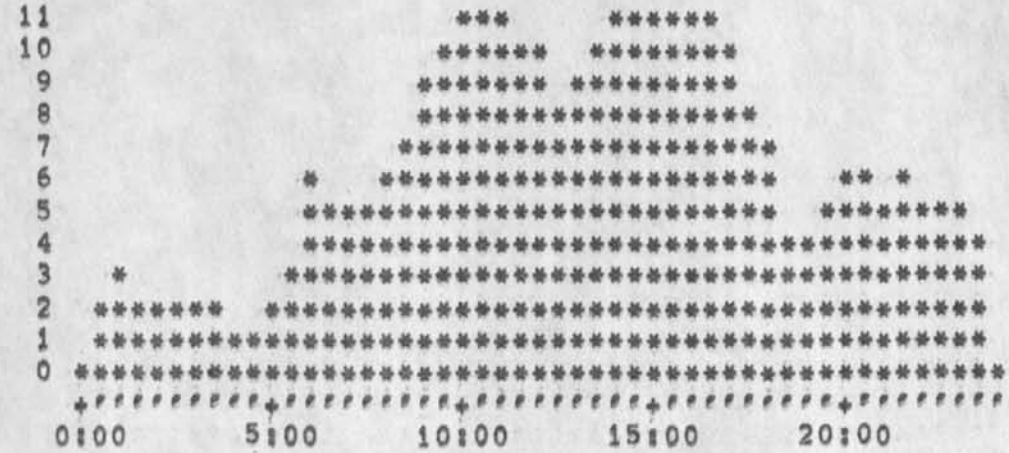
TIME PLOT OF AVERAGE PER CENT OF CPU TIME CHARGED TO USER ACCOUNTS
FOR WEEK OF 8/18/74
x axis labeled in units of hr:min, xunit = 30 minutes 3



3a

TIME PLOT OF AVERAGE NUMBER OF USERS FOR WEEK OF 8/18/74
x axis labeled in units of hr:min, xunit = 30 minutes 4

Superwatch Average Graphs for Week of 8/18/74



4a

TIME PLOT OF AVERAGE NUMBER OF NETWORK USERS FOR WEEK OF 8/18/74
x axis labeled in units of hr:min, xunit = 30 minutes

5

5a

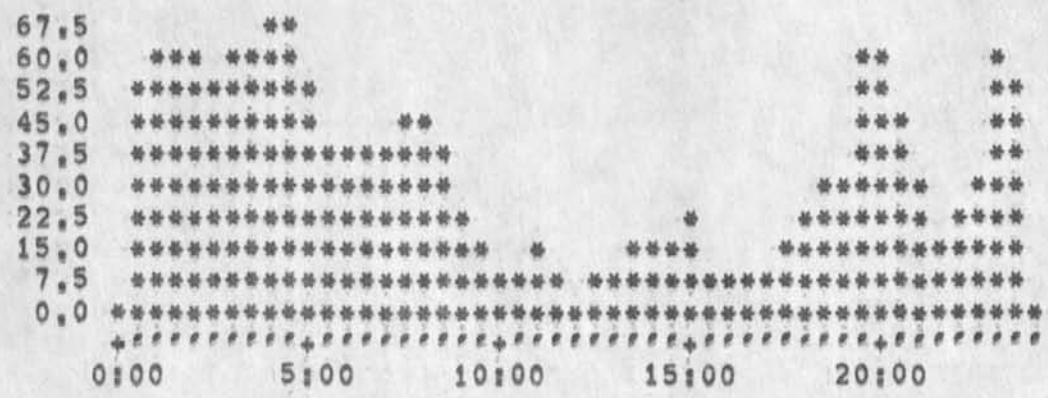
SRL 16-SEP-74 10:06 23978

Superwatch Average Graphs for Week of 8/25/74

(J23978) 16-SEP-74 10:06;;;; Title: Author(s): Susan R. Lee/SRL;
Distribution: /JCN([INFO-ONLY]) RWW([INFO-ONLY]) DCE([
INFO-ONLY]) JCP([INFO-ONLY]) DVN([INFO-ONLY]) JAKE([
INFO-ONLY]) DLS([INFO-ONLY]) DSM([INFO-ONLY]) KJM([INFO-ONLY
]) ; Sub-Collections: SRI-ARC; Clerk: SRL; Origin: < LEE,
WEEK8/25GRAPHS,NLS;2, >, 16-SEP-74 10:04 SRL
;;;;<LEE>WEEK8/25GRAPHS,NLS;1, 4-SEP-74 11:55 SRL ;####;

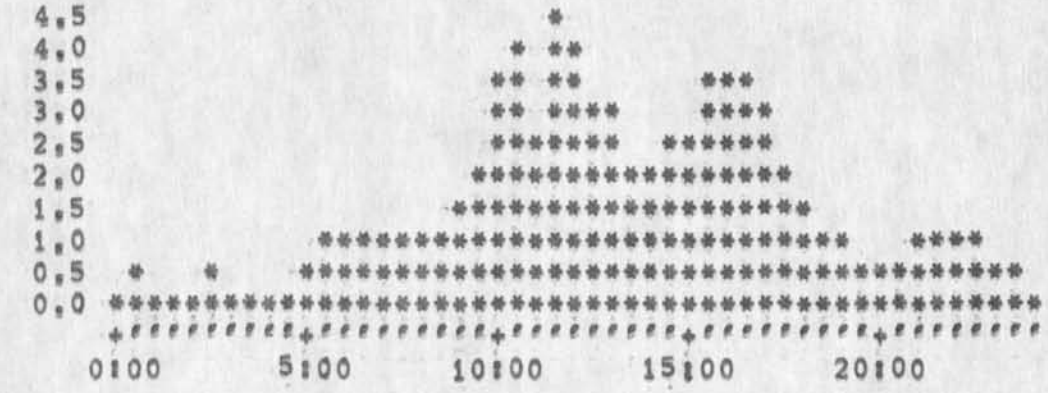
Superwatch Average Graphs for Week of 8/25/74

TIME PLOT OF AVERAGE IDLE TIME FOR WEEK OF 8/25/74
x axis labeled in units of hr:min, xunit = 30 minutes 1



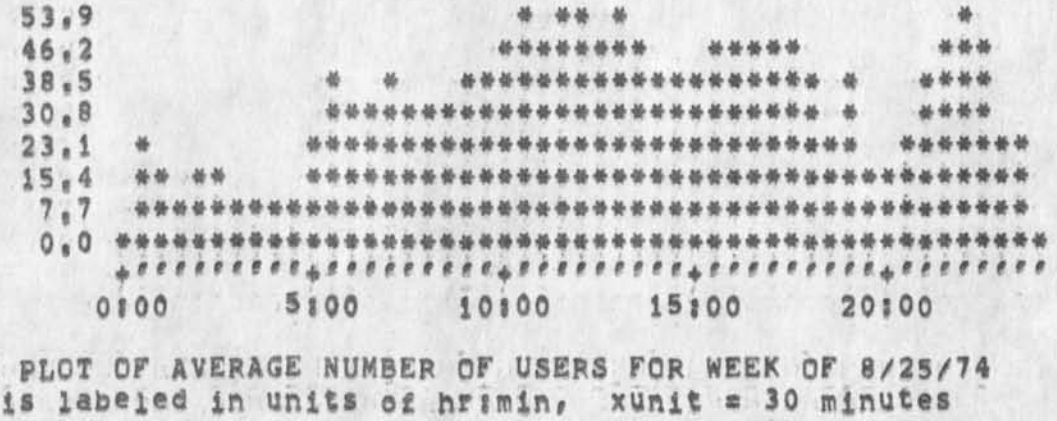
1a

TIME PLOT OF AVERAGE NUMBER OF GO JOBS FOR WEEK OF 8/25/74
x axis labeled in units of hr:min, xunit = 30 minutes 2



2a

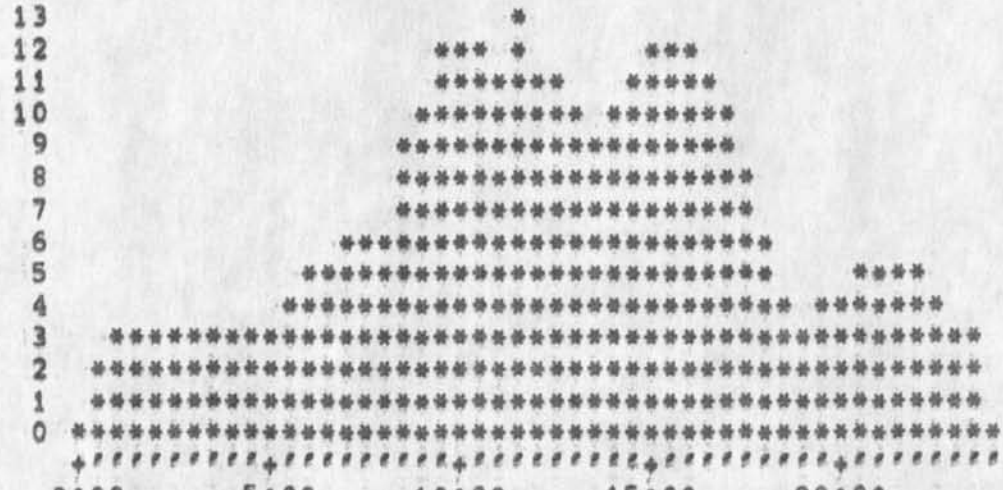
TIME PLOT OF AVERAGE PER CENT OF CPU TIME CHARGED TO USER ACCOUNTS
FOR WEEK OF 8/25/74
x axis labeled in units of hr:min, xunit = 30 minutes 3



3a

TIME PLOT OF AVERAGE NUMBER OF USERS FOR WEEK OF 8/25/74
x axis labeled in units of hr:min, xunit = 30 minutes 4

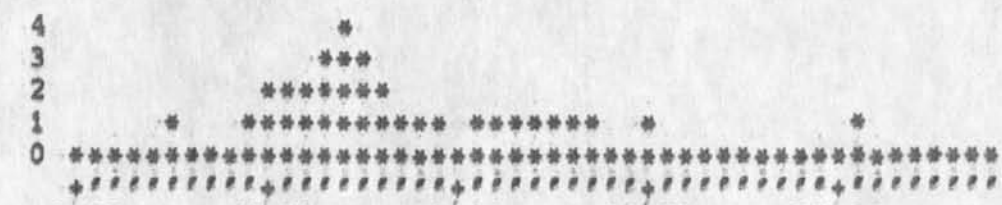
Superwatch Average Graphs for Week of 8/25/74



4a

TIME PLOT OF AVERAGE NUMBER OF NETWORK USERS FOR WEEK OF 8/25/74
x axis labeled in units of hr:min, xunit = 30 minutes

5



5a

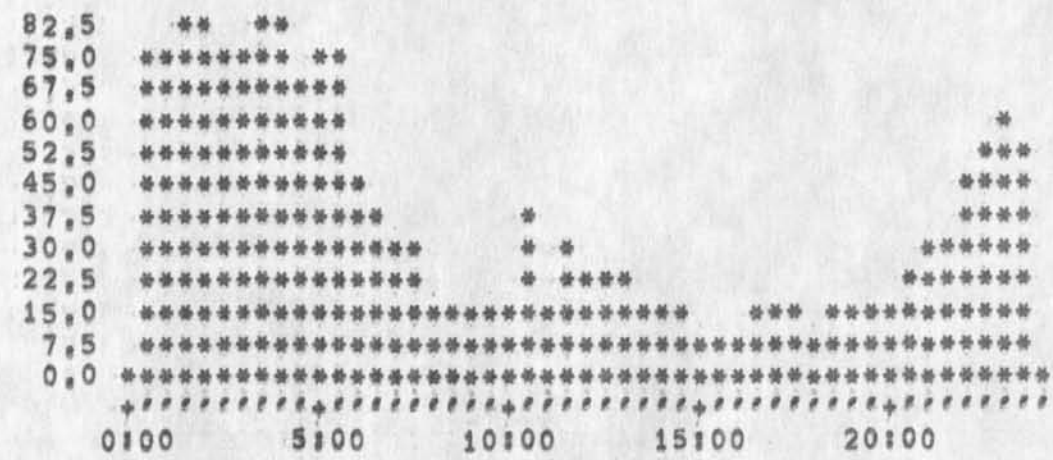
SRL 16-SEP-74 12:13 23979

Superwatch Average Graphs for Week of 9/1/74

(J23979) 16-SEP-74 12:13;;;; Title: Author(s): Susan R. Lee/SRL;
Distribution: /JCN([INFO-ONLY]) RWW([INFO-ONLY]) DCE([
INFO-ONLY]) JCP([INFO-ONLY]) DVN([INFO-ONLY]) JAKE([
INFO-ONLY]) DLS([INFO-ONLY]) DSM([INFO-ONLY]) KJM([INFO-ONLY
]) ; Sub-Collections: SRI-ARC; Clerk: SRL; Origin: < LEE,
WEEK9/1GRAPHS.NLS;2, >, 16-SEP-74 12:11 SRL
;;;;<LEE>WEEK9/1GRAPHS.NLS;1, 16-SEP-74 12:01 SRL ;####;

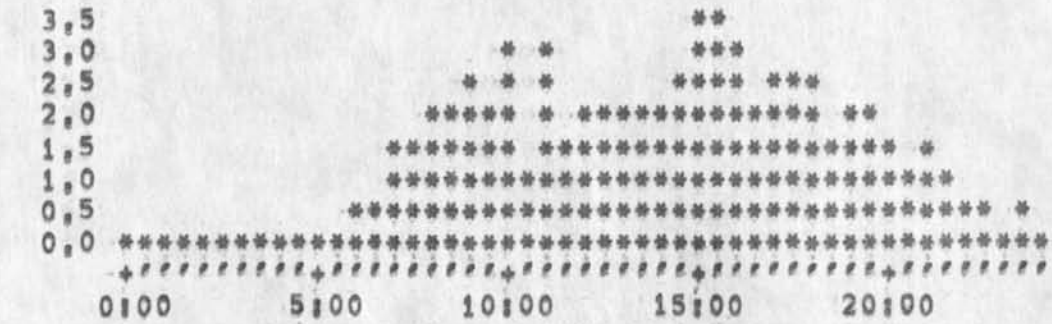
Superwatch Average Graphs for Week of 9/1/74

TIME PLOT OF AVERAGE IDLE TIME FOR WEEK OF 9/1/74
x axis labeled in units of hr:min, xunit = 30 minutes 1



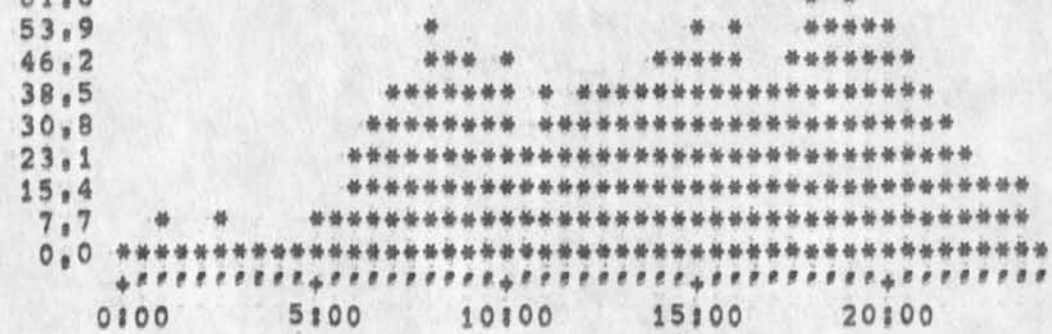
1a

TIME PLOT OF AVERAGE NUMBER OF GO JOBS FOR WEEK OF 9/1/74
x axis labeled in units of hr:min, xunit = 30 minutes 2



2a

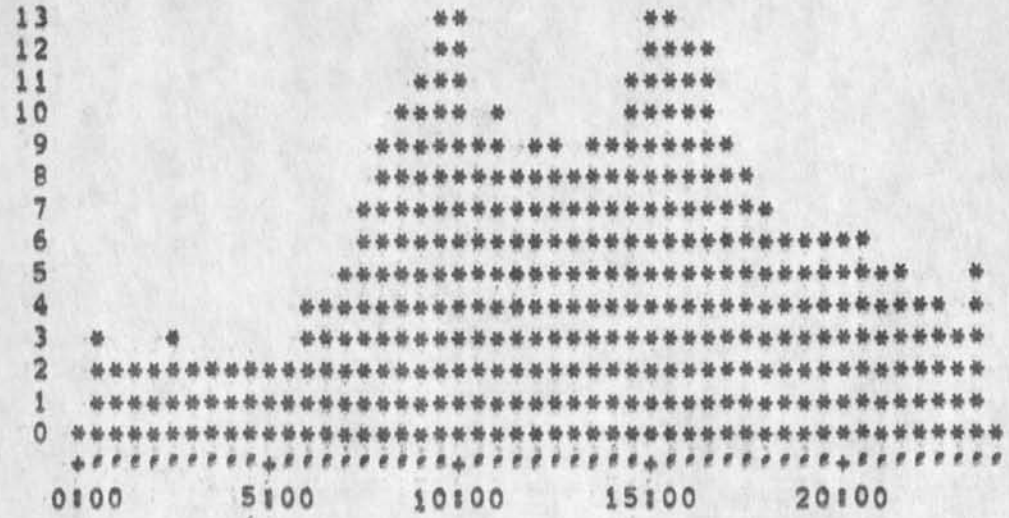
TIME PLOT OF AVERAGE PER CENT OF CPU TIME CHARGED TO USER ACCOUNTS
FOR WEEK OF 9/1/74
x axis labeled in units of hr:min, xunit = 30 minutes 3



3a

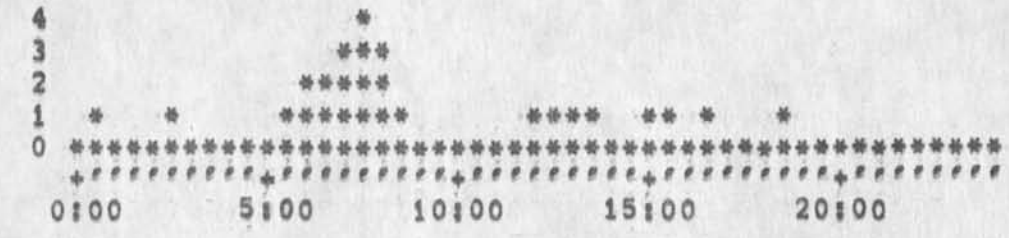
Superwatch Average Graphs for Week of 9/1/74

TIME PLOT OF AVERAGE NUMBER OF USERS FOR WEEK OF 9/1/74
x axis labeled in units of hr:min, xunit = 30 minutes 4



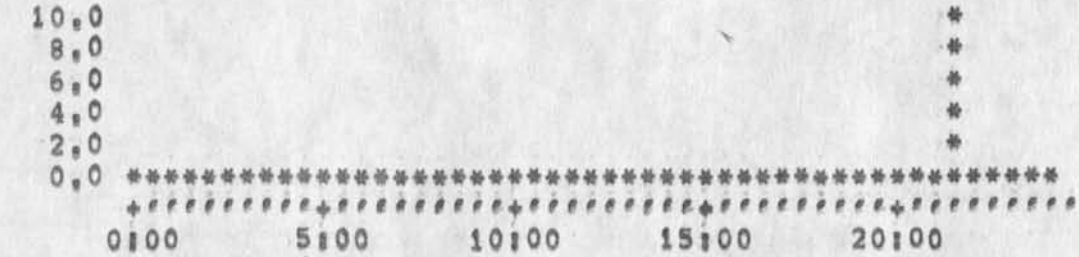
4a

TIME PLOT OF AVERAGE NUMBER OF NETWORK USERS FOR WEEK OF 9/1/74
x axis labeled in units of hr:min, xunit = 30 minutes 5



5a

TIME PLOT OF AVERAGE PER CENT OF SYSTEM USED IN OLDDNLS FOR WEEK OF 9/1/74
x axis labeled in units of hr:min, xunit = 30 minutes 6



6a

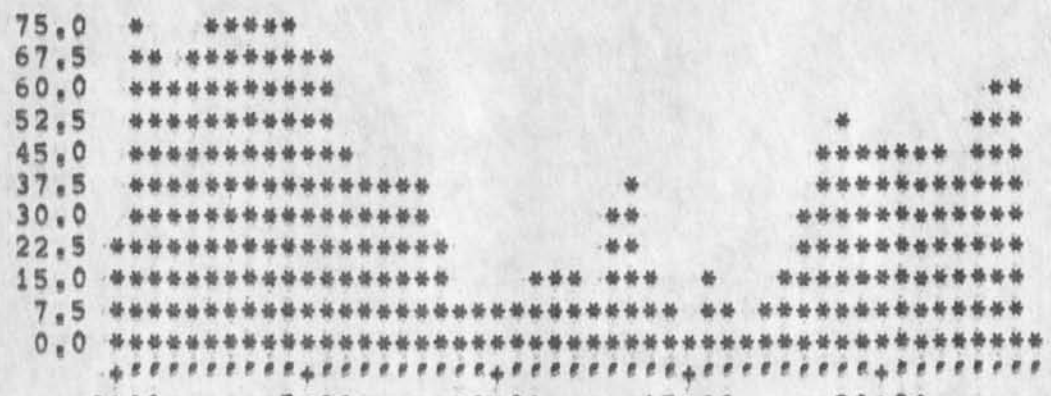
SRL 16-SEP-74 12:27 23980

Superwatch Average Graphs for Week of 9/8/74

(J23980) 16-SEP-74 12:27;;;; Title: Author(s): Susan R. Lee/SRL;
Distribution: /JCN([INFO-ONLY]) RWW([INFO-ONLY]) DCE([
INFO-ONLY]) JCP([INFO-ONLY]) DVN([INFO-ONLY]) JAKE([
INFO-ONLY]) DLS([INFO-ONLY]) DSM([INFO-ONLY]) KJM([INFO-ONLY
]) ; Sub-Collections: SRI-ARC; Clerk: SRL; Origin: < LEE,
WEEK9/8GRAPHS.NLS;2, >, 16-SEP-74 12:21 SRL
;;;;<LEE>WEEK9/8GRAPHS.NLS;1, 16-SEP-74 10:18 SRL ;####;

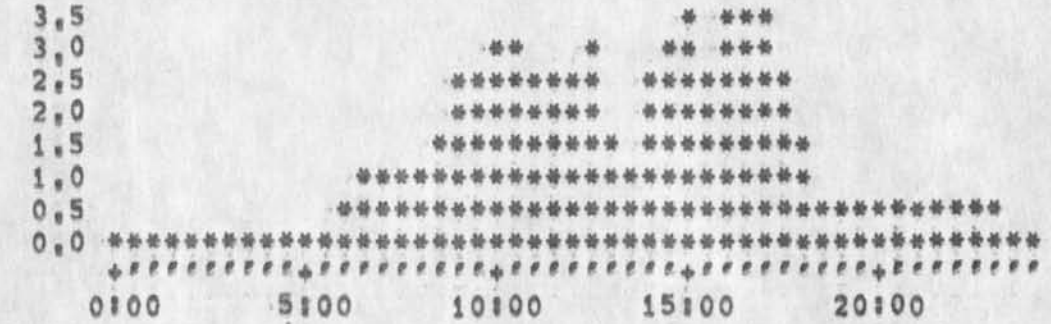
Superwatch Average Graphs for Week of 9/8/74

TIME PLOT OF AVERAGE IDLE TIME FOR WEEK OF 9/8/74
x axis labeled in units of hr:min, xunit = 30 minutes 1



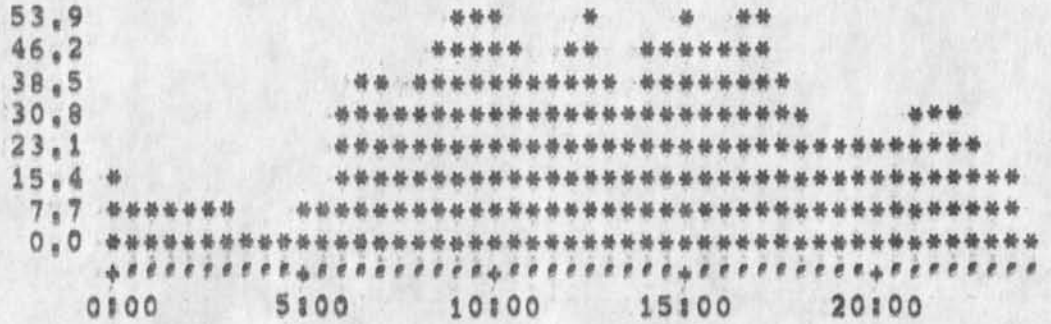
1a

TIME PLOT OF AVERAGE NUMBER OF GO JOBS FOR WEEK OF 9/8/74
x axis labeled in units of hr:min, xunit = 30 minutes 2



2a

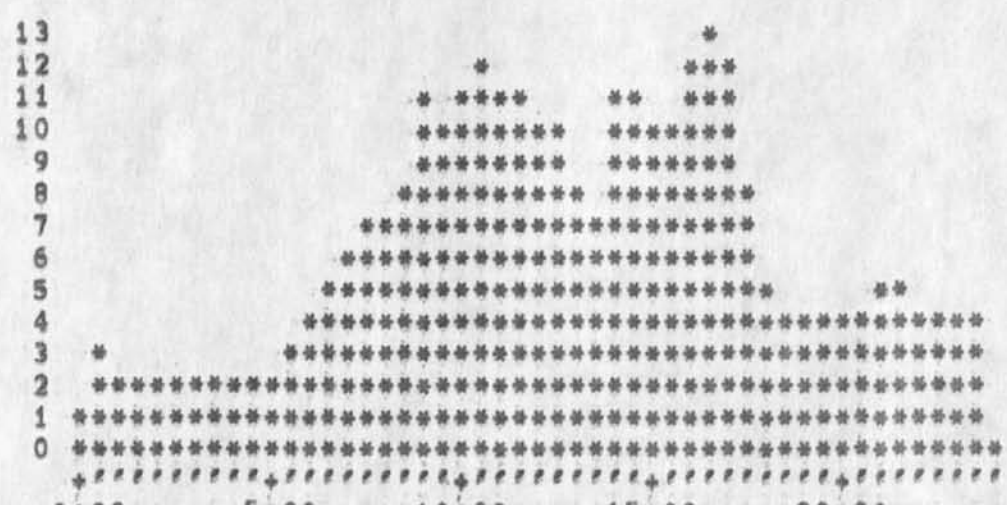
TIME PLOT OF AVERAGE PER CENT OF CPU TIME CHARGED TO USER ACCOUNTS
FOR WEEK OF 9/8/74
x axis labeled in units of hr:min, xunit = 30 minutes 3



3a

TIME PLOT OF AVERAGE NUMBER OF USERS FOR WEEK OF 9/8/74
x axis labeled in units of hr:min, xunit = 30 minutes 4

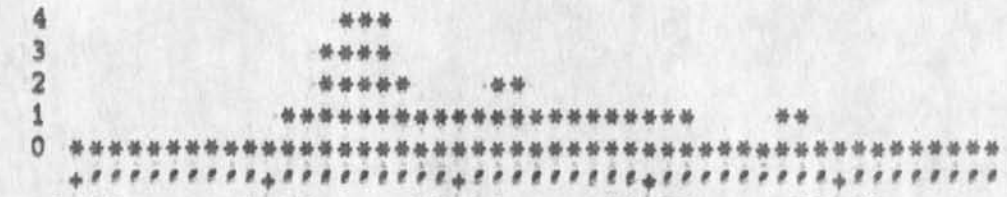
Superwatch Average Graphs for Week of 9/8/74



4a

TIME PLOT OF AVERAGE NUMBER OF NETWORK USERS FOR WEEK OF 9/8/74
x axis labeled in units of hr:min, xunit = 30 minutes

5



5a

CHI 16-SEP-74 12:39 23981

Sendmail and items under 2000 characters in length.

(J23981) 16-SEP-74 12:39;;;; Title: Author(s): Charles H. Irby/CHI;
Distribution: /KIRK([INFO-ONLY]) FDBK([INFO-ONLY]) ;
Sub-Collections: SRI-ARC; Clerk: CHI;

Sendmail and items under 2000 characters in length.

The bug that causes a file to always be created has been fixed in xnls. If a bugged entity consists of more than one statement, ten it will still become a file (this awaits harvey and dave hopper bringing up the long overdue new delevery process). 1

meeting to work out procedures to get the total system more solid

(J23982) 17-SEP-74 08:50;;;; Title: Author(s): Richard W. Watson/RWW; Distribution: /CHI([ACTION]) MEH([ACTION]) DIA([ACTION]) KEV([ACTION]) DCE([ACTION]) JCN([ACTION]) JCP([ACTION]) ; Sub-Collections: SRI-ARC; Clerk: RWW;

RWW 17-SEP-74 08:50 23982

meeting to work out procedures to get the total system more solid

Doug, is finding out first hand all the hassles of running DNLS from DC Terminal, Line Processor,Phone Line, TIP, Tenex, NLS and ARPANET all in the chain. The system is not really solid nor do we seem to have a well worked out set of procedures for isolating where a problem is. Doug has placed responsibility on Development to work such procedures out and to get the system solid. Therefore on Monday when Roddney gets back there will be a meeting at 3:00 of Martin, Roddney, Charles, Don, Ken, Jeff and anyone else who you think relevant to try to get the situation under control. Doug thinks the situation is serious and I agree. 1

RWW 17-SEP-74 09:00 23983

Split in Responsibility for the System Chain

(J23983) 17-SEP-74 09:00;;;; Title: Author(s): Richard W. Watson/RWW; Distribution: /DCE([ACTION]) JCN([ACTION]) MEH([INFO-ONLY]) CHI([INFO-ONLY]) KEV([INFO-ONLY]) DIA([INFO-ONLY]) ; Sub-Collections: SRI-ARC; Clerk: RWW;

RWW 17-SEP-74 09:00 23983

Split in Responsibility for the System Chain

When the Utility was set up we discussed the long chain of different vendors systems etc between a user and NLS and said that one of the jobs of the non existent Utility manager was to tae responsibility for he whole chain and see that it works. I still believe that is the right place for ultimate responsibility for the chain. Developmens job should be to help get a set of procedures set up to place responsibility if possible (its not clear to me that it will be a simple or souable problem) as the chain is very long. Anyway it is probably Martin at this point who has to be able to deal with problems and make whateve decisions are needed. I think Rodney may have to spend more time in DC until everybody feels comfortable. 1

JEW 17-SEP-74 16:54 23985

Inflation

(J23985) 17-SEP-74 16:54;;;; Title: Author(s): James E. (Jim) White/JEW; Distribution: /SRI-ARC([INFO-ONLY]) ; Sub-Collections: SRI-ARC; Clerk: JEW;

JEW 17-SEP-74 16:54 23985

Inflation

Eight hours didn't buy me very much today. I think NSW is in trouble unless reliability goes up and the load average goes down.
--Jim White

1

NDM 17-SEP-74 17:58 23986

User Programs to be maintained in NLS-8

(J23986) 17-SEP-74 17:58;;;; Title: Author(s): N. Dean Meyer/NDM;
Distribution: /RWW([ACTION]) SRI-ARC([INFO-ONLY]) ;
Sub-Collections: SRI-ARC; Clerk: NDM; Origin: < MEYER,
UP,NLS;1, >, 17-SEP-74 17:56 NDM ;;;;####;

User Programs to be maintained in NLS-8

ARC-Applications (as represented by JCN, NDM, RLL, JHB) feels that the following user programs should remain available under NLS-8: 1

program	estimated conversion time, hours (0=done)	comments	
			1a
First piority:			1b
address	0	should also list telephone number	1b1
deldir	1		1b2
delsp	1		1b3
format	5		1b4
index run	5	problems here to be considered in long	1b5
message	0		1b6
inseqh command	1	should be option in Copy Sequential	1b7
jform3	0		1b8
letter	0		1b9
toc	2		1b10
Second piority:			1c
addtext	2		1c1
append	2		1c2
makeref	6	problems with protected catalogs	1c3
sendmes message?	4	sendmail as front-end? or put in	1c4
showdir	1		1c5
sortnocase	0		1c6
sortrev	0		1c7
Third piority:			1d

User Programs to be maintained in NLS-8

delname	1		1d1
lowercase program	1	could be part of generalized fix-up program	1d2
sublist	1		1d3
notabs	2		1d4
wordcount	1		1d5

KIRK 18-SEP-74 01:17 23987

Jump to File Named command

(J23987) 18-SEP-74 01:17;;;; Title: Author(s): Kirk E. Kelley/KIRK;
Distribution: /JMB([INFO-ONLY]) DVN([INFO-ONLY]) ;
Sub-Collections: SRI-ARC; Clerk: KIRK;

KIRK 18-SEP-74 01:17 23987

Jump to File Named command

CHI has added a Jump to File Named comand to nls. This is mostly at DCE's request. It is quite similar to the jump to name command and does not require that the file name be a link. If the file name is in a link, the rest of thelink (including the directory) is ignored. This command will accept TENEX file name syntax, as will links. 1

JML 18-SEP-74 10:11 23989

Phone Log, Walt Scacchi, U. C. Irvine

(J23989) 18-SEP-74 10:11;;;; Title: Author(s): Jeanne M. Leavitt/JML; Distribution: /JAKE([ACTION]) ; Sub-Collections: SRI-ARC; Clerk: MEJ; Origin: < LEAVITT, IRVINE,NLS;6, >, 18-SEP-74 08:56 JML ;;;;####;

JML 18-SEP-74 10:11 23989

Phone Log, Walt Scacchi, U. C. Irvine

On 9-9 a Walt Scacchi of U. C. Irvine's Public Policy Research Organization called. He had been using an old NIC document, 14228, dated 6 Feb. 73, to try to get into the Guest account. I explained to him that the guest account no longer existed and tried to find out what he wanted to do. Basically they're looking for a system to tie into. To that end they want to try various systems out to see if they suit their particular needs. He told me their initial interest lay in the direction of a bibliographic storage and retrieval system, one capable of handling at the outset 10,000 items, and gradually growing to 150,000; however, they are very much intrigued by NLS's text processing capabilities and what he described as its "strong analytic capabilities". My first impulse was to tell him he seemed to be barking up the wrong tree, but who am I to say. At his insistence I explained to him that the only way he would be able to use nls on a non-cost basis was with the one NIC slot at office-1, and permission to use that slot would only be able to come through Jake or Jim Norton, and at that it would probably be limited to off hours. None of this seemed to phase him so he asked that I explain his situation and have one of you get back in touch with him as soon as possible. This was all in the middle of Architects' Week so I told him the earliest was this Monday. 1

His phone number is (714) 833-5449 2

NDM 18-SEP-74 10:36 23990

User Program: Deletes extra spaces in statements.

(J23990) 18-SEP-74 10:36;;;; Title: Author(s): N. Dean Meyer/NDM;
Distribution: /RMS2([INFO-ONLY]) IMM([INFO-ONLY]) JCN([
INFO-ONLY]) JHB([INFO-ONLY]) ; Sub-Collections: SRI-ARC; Clerk:
NDM; Origin: < MEYER, FIX,NLS;1, >, 17-SEP-74 12:42 NDM
;;;;####;

NDM 18-SEP-74 10:36 23990

User Program: Deletes extra spaces in statements.

Content analyzer. Leaves 2 spaces after .!?! else one, except in lines beginning with space (intentionally formatted).

NDM 18-SEP-74 10:36 23990

User Program: Deletes extra spaces in statements.

```
FILE fixsp % (110,) (meyer,fixsp.ca,) %                                    1
   DECLARE TEXT POINTER 1f, 1e, ptr1, ptr2 ;                               1a
   (fixsp) PROCEDURE ;                                                     1b
      FIND "1f SE(1f) "1f ;                                                1b1
      LOOP                                                                 1b2
         BEGIN                                                             1b2a
         IF NOT FIND 1f < "1e CH [EOL/ENDCHR] "1f THEN RETURN(FALSE)
         ;                                                                 1b2b
         IF FIND 1f > sEOL SP THEN REPEAT LOOP; %line centered%            1b2c
         FIND 1e "ptr1 ;                                                   1b2d
         LOOP IF FIND ptr1 < [2sSP] "ptr1 > sSP "ptr2                      1b2e
            THEN IF FIND ptr1 < ('.,/'?/'!)                                1b2e1
               THEN ST ptr1 ptr2 _ SP, SP  %between sentences%             1b2e1a
               ELSE ST ptr1 ptr2 _ SP      %between visibles%              1b2e1b
            ELSE EXIT LOOP ;                                               1b2e2
         END;                                                              1b2f
      END.                                                                 1b3
   FINISH fixsp                                                            1c
```

JAKE 18-SEP-74 10:52 23991

Upgrading programs for NLS Version 8

(J23991) 18-SEP-74 10:52;;;; Title: Author(s): Elizabeth J. (Jake) Feinler/JAKE; Distribution: /JCN([ACTION]) RWW([ACTION]) CHI([ACTION]) DCE([INFO-ONLY]) NDM([INFO-ONLY]) JDH([INFO-ONLY]) ; Sub-Collections: SRI-ARC; Clerk: JAKE;

JAKE 18-SEP-74 10:52 23991

Upgrading programs for NLS Version 8

I would like to know what the procedure is to have current programs upgraded so that they run in the new NLS. Many of my maintenance programs do not now run in the new NLS and I was told to run them in old NLS until they could be upgraded. If old nls is going away soon this will create quite a problem. Can I please have some guidelines as to how you would like to have this handled. Thanks, Jake 1

RWW 18-SEP-74 18:36 23992

Initial Reaction to User Program List from Applications

(J23992) 18-SEP-74 18:36;;;; Title: Author(s): Richard W. Watson/RWW; Distribution: /SRI-ARC([INFO-ONLY]) ; Sub-Collections: SRI-ARC; Clerk: RWW;

RWW 18-SEP-74 18:36 23992

Initial Reaction to User Program List from Applications

In looking over the list of user programs that applications wants brought NLS 8 and I assume maintained across system changes it is clear that they fall into easily identifiable categories and represent peoples attempts to remove holes in the standard system. Basically it is my guess that they should be either made into commands in subsystems to be suggested below or added to existing commands or subsystems and no longer maintained as user programs. Dave, Charles how much extra work over Deans estimates would this be?. One might still argue whether or not certain of the suggested capabilities are really that important or should be maintained by development. What do Dev people think? 1

An initial stab is made here as to where they might belong. 2

A Document Makers Subsystem 2a

Deldir, format, showdir (one class) makref toc index workcount (a second class) 2a1

Sort 2b

The sort command should allow sort keys etc to be specified as options 2b1

sortnocase sortrev 2b1a

Letter 2c

I can imagine a growing letter writing aids subsystem 2c1

letter 2c2

Editing Expansion 2d

notbs sublist lowcase delname append addtext delsp 2d1

Additions to Ident sYstem in sendmail subsYstem 2e

address 2e1

Additions to Sendmail 2f

message jorm3(the new default dellivey format?) sendmes 2f1

uAn improvement in input sequential 2f2

inseqh 2f2a

JAKE 19-SEP-74 08:48 23993

Incorporating user-progs into NLS 8

(J23993) 19-SEP-74 08:48;;;; Title: Author(s): Elizabeth J. (Jake) Feinler/JAKE; Distribution: /SRI-ARC([INFO-ONLY]) ; sub-Collections: SRI-ARC; Clerk: JAKE;

JAKE 19-SEP-74 08:48 23993

Incorporating user-progs into NLS 8

Basically I think Dick has a good point that many current user progs could be incorporated into the system. On the other hand I would like to see some of them such as makeref and document formatting be carefully thought out by a design group with some preview interaction by users. It seems like these cases should wait for NLS 9 given the current burden on programmers. 1

JCP 19-SEP-74 11:17 23994

Journal Indices

(J23994) 19-SEP-74 11:17;;;; Title: Author(s): Jeffrey C. Peters/JCP; Distribution: /SRI-ARC([INFO-ONLY]) ; Sub-Collections: SRI-ARC; Clerk: JCP;

JCP 19-SEP-74 11:17 23994

Journal Indices

In response to repeated questions, we are indeed maintaining the online Journal Indices. The most recent entries are about 1 month old at this time an I hope to get them current a little later. When I get them caught up here, they will also be moved to Office-1. If you want to find out how to reach these indices, jump to the following link: (userguides, locator, 5) - Jeff 1