PART I: OVERVIEW OF THE NATIONAL SOFTWARE WORKS PROJECT 2

Introduction 2a

Software production in the DoD is estimated to cost over $3 Billion per year, (Ref. 1) and dominates the schedule of development of almost all computer systems. Yet programming remains a loosely controlled manual process, with little automated assistance. There are numerous reasons for this unsatisfactory state of affairs, but probably the most important is the fact that tools which can materially aid programmers, analysts and their managers are inherently expensive to develop, and typically require computers much larger than those required to run finished programs. Since most programmers are constrained to use the same computer for both development and operations, only the simplest and widespread tools are developed for each computer, and new tool development is inhibited. 2a1

Computer networks, and an order of magnitude decrease in the cost of on-line storage, provide an opportunity to attack the software production problem in a more complete way than ever before. In the Summer of 1973, the Defense Advanced Research Projects Agency(ARPA) organized discussions among a number of professionals from industry, the Services and universities. The general notion emerged of a software factory implemented on a computer network, with a coherent collection of tools which would expand and become more powerful over time. 2a2

Further discussions with the Services in the Fall and Winter of 1973-74 refined the concept, and led ARPA to form a joint program with the Air Force Data Automation Agency(AFDAA) to implement the first version of a distributed software factory, to be called the National Software Works (NSW). Air Force Systems Command (AFSC) joined the project in the Summer of 1974 through its computer science research organization at Rome Air Development Center. Discussions are continuing with the Army and the Navy, and it seems likely that a coordinated tri-service effort will evolve. The National Software Works can potentially provide for DOD-wide utilization of capabilities which otherwise will be created ad hoc for each new software development program, or worse, not be available at all. 2a3

Background and Technical Need 2b

The Software Problem (Excerpted from Ref. 2) 2b1

Over the last ten years, there has been a radical shift in the balance of hardware and software costs. The cost of

computing is clearly dominated by the cost of software. Since software is often a critical component in large systems, overruns in delivery time or serious flaws can have hidden costs and penalties that exceed the direct hardware and software development costs. 2b1a

Demands for software production are increasing in volume and complexity, but progress in software technology has been slow. The demands have clearly outstripped the state-of-the-art, with very costly results. Cost overruns on software development projects are legendary. Software is seldom delivered on time. There is much waste in programming and computing, resulting from poor matching of software and hardware. Incompatibility between computers results in costly reprogramming or an inability to take advantage of the reduced computing costs of new hardware. The maintenance costs for old software products may be an order of magnitude larger than production cost, due to poor original design and production. 2b1b

The high direct and indirect costs of software set an effective practical limit to the complexity and scale of realizable systems. A major reduction in software costs (including the costs resulting from flaws) could have a great impact on the practical capability of logistic, avionic, tactical, communication, and other vital systems. 2b1c

The rapidly decreasing costs of computation resulting from new technological advances and the rapid growth in computer networks will, together and separately, cause a large expansion in the population of computer users and a large increase in the variety of applications. The threshold of economic feasibility is dropping for many systems, and awareness of how to employ computations is spreading to many sectors where computing is not a present activity. This will result not only in more computer usage but also in the need for much more software. 2b1d

In addition, major changes are occurring in the character of computing. Batch-mode processing currently dominates computing, but there is a strong trend toward on-line computing of ever-increasing scale. Requirements for complex real-time processing in such areas as tactical systems are also growing. The present software art is poorly matched to the current methods and levels of computation, and as these modes grow in importance, software costs will escalate. The "learning" costs incurred as the art strives to meet new kinds of demands will be high indeed. 2b1e

Great advances are needed in making computers more natural for people to use and in finding the right level and character of man-machine interaction. The present software art is only at the beginning of such capability. Some forms of man-machine communication will require major increases in software complexity, to match human sensory and intellectual power. 2b1f

Finally, as computing becomes more widespread, the problem of tracking users' requirements will become acute. Keeping up with changing requirements may already be the biggest source of DoD software costs, not only in the maintenance phase, but also during the development phase. In the future, these costs will increase as more DoD functions are partially or completely automated. 2b1g

Why So Few Tools? 2b2

The tools used to develop software do not reflect software's relative importance in determining the cost, reliability, and delivery schedule of the total system. In most billion dollar industries, a substantial investment is accumulated in supportive tools. The development of such tools is difficult for labor intensive activities like software production, where each product is somewhat unique; but the real barrier to the development of adequate tools to support software production has been the requirement that the tools be reimplemented for each new kind of hardware. Converting development tools to run on different hardware is usually more difficult than converting an applications system. Since a prime use of software tools is to shield the applications programmers from the details of the computer hardware, the logic of the tool must embody specific knowledge of the hardware characteristics if the results are to be efficient. In the face of conversion costs, some valuable tools have been lost. For example, there were satisfactory solutions to the problems of round-off, overflow, and underflow in numerical computations for the IBM 7094. That was 1966, but the problems are still recurring in newer, and in theory more sophisticated, computer systems. 2b2a

For planning purposes, the Air Force uses six years as the economic life of computer hardware. That means that almost all applications systems development must be completed in the first year or two after a system is installed if the development costs are to be recovered. Tools which are developed after the hardware is delivered will also come after the programming staff has finally become accustomed to the new system and developed standard procedures for using

it. Since new tools will be completed too late to help with the bulk of the applications systems, and constant retraining is something operational organizations can ill afford in any case, there is little incentive for people outside of the software R&D community to build tools. 2b2b

A related problem is the fact that machines are usually sized for their production requirements, not their development ones. Hence, they typically do not contain enough mass storage for the files that would be required in an on-line environment, nor enough memory to support both the code being developed and the tools used during that development. Additionally, access to the system is limited by the priorities of the production work load. A little recognized fact is that the tradeoff between manhours and machine resources is vastly different during development than during production. The CCIP-85 study (Reference 3) has shown that development costs increase exponentially as the machine approaches saturation. 2b2c

Despite these problems, the inventory of support software has been gradually expanding. Among the most widely used software tools are compilers, operating systems, time-sharing executives, file systems, program librarians, and interactive editors. Virtually all multi-programming operating systems have attempted to create a suitable programming environment by providing a set of tools. Some merely provide a library from which tools can be selected one at a time by the programmer. Others, like Multics, CP-67, VS-370, and TENEX, have provided an on-line environment for program building and debugging. 2b2d

These systems have not been as productive as they could have been, because there are at present no interfacing standards which assure that tools can be used together effectively. Non-integrated, tool-at-a-time operation places too great a load on the programmer to specify exactly what operations are to occur. This problem is particularly acute since tools often have command language idiosyncracies. On the other hand, if the tools supporting a programming language are tightly integrated, then it is at present impossible to access them from other languages. For example, the APL environment is completely isolated from the rest of its host IBM 360 or 370. Thus, tools may have to be duplicated for each language supported on a hardware system, as well as for all the different kinds of hardware systems. 2b2e

Origins of the National Software Works Program 2b3

Recent technological advances make it possible to overcome the barriers which have prevented the accumulation of a collection of powerful tools to support the software development process. The costs of both processing and on-line storage are dropping rapidly, so it will soon be feasible to have all programmers working on-line. Experiments like the Programmer's Interface have shown that many software tools are language independent or only slightly language dependent. Experiments using the ARPANET have shown that programs running in several machines can cooperate and appear to the user as a single system. Such cooperation is possible even if the host machines were built by different vendors and have significantly different architectures and operating systems. Finally, there are several examples of large time-sharing systems being used to support development environments for other kinds of hardware, in particular mini-computers. 2b3a

Thus there is strong evidence to suggest that tools running on a diverse collection of computers can be used together effectively to develop software for a variety of target machines, using a variety of languages. The key is the definition of appropriate interfacing standards. 2b3b

Recognizing the significance of the Software Problem to the DOD, and believing that these technological advances offered an opportunity to attack that problem in a much more concentrated way than ever before, ARPA/IPT held a series of meetings with software specialists from industry and government during the Summer of 1973. Among the participants were Barry Boehm(TRW), John Brown (TRW), Michael Busch(CSC), F.J. Corbito(MIT), Peter Deutsch(XEROX PARC), Jerry Feldman(Stanford), Cordell Green(Stanford), J.C.R. Licklider(MIT), Tom Lippiatt(Rand), Barbara Liskov(MIT), Richard Watson(SRI), Clark Weissman(SDC), Robert Balzer(ISI), T.E. Cheatham(Harvard), Stephen Warshall(Massachusetts Computer Associates), Stephen Crocker(ARPA/IPT), William Clark(NAVSHIPS), L/C Robert O'Keefe(USAF-ESD), Maj Harold Arthur(USAF-ESD), Norman Glick(NSA), John Mott-Smith(USAF-ESD), and Maj Zara(USAF-ESD). The result of these meetings was a report (reference 4) and a determination that the project should be jointly sponsored by a Service organization which is a major producer of operational software. 2b3c

During the Fall 1973, all three Services were presented with the National Software idea. The strongest interest was expressed by the Air Force Data Automation Agency. The Army

Computer Systems Command also assigned an officer to participate in NSW planning sessions. 2b3d

AFDAA tasked two of its components, the Air Force Data Services Center (AFDSC) and the Air Force Data Systems Design Center (AFDSDC), to participate in the project. AFDSDC is located at Gunter AFS, Alabama and is responsible for developing and maintaining standard data systems which run at over 130 AF bases throughout the world. AFDSC is located in the Pentagon, and supports the Headquarters Air Force and the Office of the Secretary of Defense. (The third organization within AFDAA, the Federal ADPE Simulation Center in Springfield, Virginia, is not currently participating.) 2b3e

The first meeting of an NSW Steering Committee was held in October 1973 and was attended by Mr. Steve Crocker (ARPA/IPT), L/C Gray Kinney (US Army Computer Systems Command), Maj Tony Baggiano and Mr Al Mayhan (AFDSDC), Maj James Lloyd and 1Lt William Carlson (AFDSC), Dr. Robert Balzer (ISI), and Mr. Steve Warshall (Massachusetts Computer Associates). 2b3f

In March 1974, a plan (Reference 5) was published for a joint AFDAA and ARPA effort to build the National Software Works. The Army Computer Systems Command determined that it did not have the funds available to actively participate at that time. The Plan was briefed to Col T.L.McGovern, AFDSDC Commander, Col E.O. Wells, AFDSC Commander, and to MG J.B. Robbins, the Commander of AFDAA. They approved the Plan, and in April 1974 MG Robbins and Dr S.J. Lukasik, the Director of ARPA, signed a Memorandum of Understanding (Reference 6) to carry out the development. 2b3g

Rome Air Development Center (RADC) joined the program during the Summer 1974. RADC, located at Griffis AFB, New York, is the component of the Air Force Systems Command which is responsible for advanced computer science research. They are jointly funding the development, and are also serving as the ARPA Agent for NSW contracts. 2b3h

National Software Works Design Concepts 2c

Overview 2c1

The National Software Works will be a software development environment on a very large scale. It will be built on a computer network to reach a wide user community, and will integrate a continually growing collection of specialized

services into a coherent system to support the development of software for a variety of diverse hardware. A typical terminal session may involve operations on several different machines. 2c1a

Consider, for example, the construction of a standard Air Force software system for the Burroughs B3500 using the COBOL language. Programmers may want to use the NLS editor on a PDP-10 to enter their source programs and to prepare their documentation. To reduce the cost of on-line storage, some of the files may be stored elsewhere on one of the new trillion bit storage devices like the Ampex TBM, which offers on-line storage at a cost of about a dollar per megabit per year. A Burroughs B6700 might be used for preliminary syntax checking. Interactive debugging at the source code level might then be done either on the B3500 or on a B4700 (a larger, faster version of the same machine.) Perhaps the best test data generator runs on an IBM 370. Finally, the software should be tested on a Burroughs B3500. The machine on which a tool runs has come to be called a Tool Bearing Host (TBH). The essence of the National Software Works idea is to make the best possible tools available by decoupling the selection of Tool Bearing Host hardware from the selection of production hardware. 2c1b

Initially, the NSW will be built on the ARPANET, which interconnects fifty computers, distributed over the United States, London, and Hawaii. An incomplete list of the operating systems at various hosts includes TENEX, ITS, and 1050 for the PDP-10, ANTS and ELF for the PDP-11, Multics for the HoneyWell 6180, MCP for the Burroughs B6700, and variations of OS, VS, CMS, and TSS for IBM 360s and 370s. Its user community includes many experienced researchers working on ARPA supported projects. These researchers will provide constructive criticism, and the results of their research will become directly available to DoD personnel through the NSW. 2c1c

At present, the ARPANET is merely a communications system for interconnecting independent facilities. Each machine is owned by a different organization. The user must have registered himself in advance at each site, have established credit, and arranged to be billed for the time he uses. He must know how to log into each machine, how to invoke services of each operating system, and how to transfer files among them. 2c1d

The NSW which will eliminate many of the problems associated with the current ARPANET. It will centralize accounting and

automatically perform host logins, tool-invocation, file access, and file movement for the user. The software system which accomplishes this is referred to as the Framework. Initially, it will run in a single network host, but eventually the Framework will be distributed across hosts. The parallelism is needed both for capacity and reliability. Rules will be defined for adding new tools to the environment. 2c1e

It is understood that some effort will be required to install tools in the NSW; that is the price of achieving standardization without discarding existing operating systems. There are no plans at present to have the Framework optimize the use of hardware resources. Tool installers and users will decide where files are to be stored, which hosts will support a particular tool, and whether whole files or partial files should be moved to perform a given operation. The Framework will help them make good decisions by simplifying the implementation of the various alternatives, and by providing feedback on costs. 2c1f

In order to provide a more consistent interface, users will access the NSW through Front-End network access machines. The access machines will know which characters require action by the tool being used, so that input characters can be collected and transmitted in blocks. The access machine will also support some local command interpretation and user prompting. Whether additional functions should take place at the user site, for example text editing, is an open issue. 2c1g

A New Capability for Project Control 2c2

A view of the NSW as a mere lash-up of tools which happen to reside on the ARPANET would be extremely short-sighted. The fact that all programmer contact with tools passes through a common communications media with immense computing resources creates an opportunity for the study--and perhaps control--of the whole process of large program creation and maintenance. 2c2a

In the production of a large software system, numerous programmers, analysts, and managers cooperate in a venture whose end product is, in some sense, a single entity. In the course of their work toward this goal, they prepare, edit, and manipulate a very large number of pieces of "text" of various types: routines in a programming language, data descriptions, structured data objects, modules of object code produced by a compiler, assemblages of such modules

linked together by a link editor, items of program documentation, and so on. 2c2b

To the degree that all of these types of text are either machine-processable or machine-producible, it is reasonable to say that they are all either prepared (and repaired) by project members or produced by "tools" by which we mean elements of support software invoked by computer specialists to operate on pieces of text. 2c2c

The number of such pieces of text which come into existence in the course of a large project can be astronomical, and even the number in some kind of active status at a particular time is likely to be huge. It ought to be clear that any absence of control over this large and shifting inventory of material is an invitation to confusion and the almost total absence of any support software for "inventory control" might have something to do with the high and uncontrolled cost of program production (and perhaps something to do with our difficulties in figuring out what we are doing wrong). 2c2d

Suppose by contrast that the total inventory of text pieces were explicitly regarded as one logically integrated data base -- the Project File -- and that some piece of support software were charged with the responsibility of managing that data base. This piece of software - for the moment, let us call it the File Manager - would of course, keep books on the contents of the Project File. These books would include not only the character and status of each item in the Project File, but also its relationship to other items in the File (that A is a later version of B, that C is the object code module corresponding to COBOL test D, and so on). 2c2e

It should be obvious that, if we have designed the books correctly and arranged matters so that they are always kept accurately and completely, they provide the data crucial to any serious attempt by management to explore or control what is happening in the project. 2c2f

It is, of course, essential to any interesting use of the project books that they always be complete and correct, and that there be no path of entry to the Project Files unguarded by the File Manager. This suggests strongly that an individual programmer's use of his tools -- at least when that use yields a non-transitory (Filed) result -- must always be reported to (and, perhaps, controlled by) the File Manager. 2c2g

To arrange matters so that this requirement is met is extremely difficult when the support software designer is confined to the resources of a particular local hardware: to keep the File Manager and its books effectively on line at all times may be insupportably expensive. Indeed if a projects development work is performed on several computers with no communication among them, it may be logically impossible to create a reasonable File Manager. Thus, it is not surprising that there has been no serious attempt to provide a facility of the sort we have described: at least the naturalness, if not the feasibility, of the idea depends on a unification and scale of computing resource found only in gigantic machines or in networks. 2c2h

A fairly powerful query system will be provided to answer questions about any filed entity: what it is, where it came from, what other entities depend on it, etc. Later we will introduce a variety of experimental tools for project control which use the File Manager's books as their primary data or use the fact of the File Manager's existence as their means of invocation (after all, the later provides a single control point "awakened" every time anything interesting happens). Here are some proposed tools: 2c2i

Project Status Reporter: This relates the present status of the files to the overall project plan (in machine-readable form), identifying bottlenecks, critical paths, etc. 2c2i1

Project Accountant: This produces reports on the frequency and cost of various patterns of activity interesting to project management. 2c2i2

Policy Enforcer: Everybody in Section A must use the same version of function X; no programmer may link up two routines until each is adjudged debugged by a section manager; no programmer may start debugging until all his code is written; no programmer may write any code for phase 2 of the project until he has written all his code for phase 1; no programmer may start writing a new routine until his last is documented. The above list of (rather inane) policies are meant to suggest a large family of more reasonable policies which might apply to some or all programmers at various phases of a project. If a plausible way of expressing such policies in machine-readable form can be developed, it is no great trick to devise a tool which is invoked by the File Manager to verify that the present action of some

programmer is consistent with policy, so that the action may be inhibited or permitted accordingly. 2c2i3

The use of such new tools by the project would of course, be optional. In any event, the research community can make use of such tools to collect the data it needs to discover what makes program development and maintenance so expensive. 2c2j

Internal Design 2c3

The software which comprises the NSW Framework can be divided into three major components which have been given the (hopefully suggestive) names: Work Manager, Foreman, and Front-End. Each of these these names refers to an aspect of the NSW which requires analysis and design, and whose result will be some set of programs. The programs will communicate with each other using a new Procedure Call Protocol (PCP). 2c3a

(Referencexxx) 2c3a1

Works Manager 2c3b

The Works Manager is the heart of the NSW. It acts somewhat like an operating system in that it accepts requests for the performance of work (requests for tool use), arranges the initiation of that work, keeps track of work in progress, does cleanup after completion of a piece of work, manages file storage, and so on. It differs sharply from a conventional operating system in that its primary function is not optimal resource allocation, but rather validation of the work request and protection of the integrity of the files. Here, both validation and protection are to be taken in a far wider sense than has heretofore been customary in the programming field. 2c3b1

The Works Manager will create and maintain an extensive catalogue of each project's inventory of filed objects. This catalogue will include both structural and historical information about each object (when was it made at whose request, by what tool, from what other filed objects; what other objects have been made using this one as data; what truths about this object have been asserted, or proved; and so on). Since the Works Manager will also have access to descriptions of the characteristics of all tools within the NSW, it can be seen that validation of a work request can be unusually exhaustive: does the tool exist, do the input files

exist, are they of the right type and status for this tool, are the files and the tools both of the right status to be used by this user at this time, and so on. 2c3b2

In the same way, any entry of a new object into the files will be supervised by the Works Manager, which will perform all updates to the catalogue implied by the existence of the new object. Thus, the idea of file integrity is expanded to include catalogue integrity, which both permits strong validation of work requests and transforms the catalogue into a powerful data base for future tools (ranging from a simple query system which answer questions about the contents of the project files to experimental tools in automatic programming which map a desired object into a best sequence of tool calls to create it -- given what objects are there already). 2c3b3

Foreman 2c3c

The Foreman is that component of the NSW concerned with taking a well-defined and fully validated request for tool use and actually getting the job done. It receives an encoded message from the Works Manager which says, in effect, "At the request of user X on host Y, execute the COBOL Compiler on Host Z, using file HENRY on the Datacomputer as input and filing the result on the Datacomputer under the name GEORGE and, when the job is finished, send me the message W." The Foreman worries about fighting the host protocols and arranging the communication so that the requested task is in fact performed. 2c3c1

While the Works Manager component clearly has as implementation some integrated family of programs resident on the NSW hosts, the structure and location of the programs which do the Foreman's job are by no means so self-evident. A portion of the work will be centralized in the NSW host, but other portions will be handled by pieces of program implemented in each Tool-Bearing Host and in each Front-End user access machine. There are some complex issues here, where best design may be different for "closed" tools, like compilers, on the one hand and "open" tools, like editors, on the other. The goal will be to establish a small number of standards, called "Tool Bearing Host Protocols", which all Tool Bearing Hosts and tools must obey, and leave each tool installer as much flexibility as possible to take advantage of special characteristics of his environment. 2c3c2

Front End 2c3d

The Front-End will normally execute in a mini-computer (initially a PDP-11) which sits between the user's terminal and the ARPANET, but the system will also run on a PDP-10 TENEX time-sharing system to support users whose terminals are connected to TIPS, TENEXs, or large Hosts. All commands to the Works Manager or to any tool must be given through the Front-End. It will provide terminal control, aid the user in command specification, parse commands, and communicate with the appropriate resource(s). 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 NSW. A user will learn to use additional functions by increasing vocabulary, not by having to learn separate "foreign" languages. When in trouble, he will invoke help or tutorial functions in a standard way. 2c3d1

The Front-End will inform the Works Manager whenever it recognizes that the user is attempting to access or create new files. Eventually, the Front-End will 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. 2c3d2

Tool builders will be able to take advantage of the Front-End's Command Language Interpreter (CLI) instead of having to develop their own user interface. The CLI will also be able to handle machine-oriented messages from NSW tools or the Works Manager and translate those messages to an appropriate man-oriented language form. It is driven by two data structures, a Command Language Grammar and a User-Profile data structure. These data structures are sent to the Front-End either by the Works Manager or by the tool system as needed. 2c3d3

A Command Meta-Language (CML) for specifying the user interface will be developed, and a compiler will be implemented to produce command language grammar data structures from the CML. Tool installers will be able to use the CML to define their user interfaces. The User-Profile will control such things as how much help or prompting a user receives when using a particular tool, what commands are available, and other information

tailoring the system to the user. The information in this data structure can be changed upon user request, or (eventually) adaptively by programs based on user behavior. Updates to the User-Profile will be reported to the Works Manager. 2c3d4

The Front-End will support CRT displays and hardcopy typewriter terminals, as well as CRTs used as full two-dimensional device, through a Terminal Controller. The Terminal Controller is a microcomputer supporting primitives for subdividing display screens into rectangular windows and for allowing the user to "point" at information displayed on a screen as arguments in commands to tools. The Front-End will also be able to handle such devices as card readers and punches, line printers, and tape drives. 2c3d5

An operating system interface module, consisting of a set of virtual operating system primitives, will be used to make the Front-End as operating system independent as possible. This will enable the Front-End to be transferred to other equipment later. The Front-End systems will be maintainable, loadable, and (symbolically) debuggable from a remote timesharing (TENEX) system. 2c3d6

Generalized Support tools 2c4

Yellow pages 2c4a

The NSW can be viewed as a library of development tools. A catalog, or yellow pages, is needed so that users can locate applicable tools when they start to work on a new kind of problem. The directory system could be based on keywords, or it could be a full text retrieval system which operates against the narrative descriptions of the tools. In the long-term, a hybrid of those two approaches will be needed. The directory should include information about a tool's cost, maintenance status, reliability (including trouble reports), who maintains it, and the existence of a user community. Tool installers should be able to determine if organizations which do not at present have access to the tool should still be made aware of its existence via the directory. 2c4a1

Software libraries. 2c4b

There are many cases when software should not be written at all -- existing code should be used. The NSW must

provide access to software libraries to facilitate their use. Ideally it should be possible to access all available libraries using a standard command language. However, in the short run, only a limited number of such libraries can be made available, and there will still be some command language idiosyncracies. 2c4b1

The two major classes of software libraries are libraries containing subroutines or components which can be used as building blocks and incorporated into the applications software, and libraries of self-contained applications systems. These two kinds of libraries will have to be handled differently, especially with regard to accounting and release (copy permission). 2c4b2

The difficult cases are the libraries of subroutines or components which the user can incorporate into the software system he is building. Access control is very difficult with such tools. Three alternatives have been identified thus far: 2c4b3

Subroutines and other components can only be used in the vendor's environment and the system will not allow you to copy a load module out without paying a stiff exit fee. 2c4b3a

The subroutine can be called across the network for debugging, and the user would again be charged a stiff exit fee. 2c4b3b

Only certified linkage tools are allowed to access the subroutine library, and the user is charged each time he builds a new object module. 2c4b3c

An effort should be made to get libraries of government-owned software installed as tools. Regardless of the rules for releasing components, a record must be kept with information about who uses the subroutine, on what machine, under what operating system and with what result. There must be a complete audit trail of all copies so that the users can be notified when the subroutine or component is modified. 2c4b4

These release control considerations identified above apply to self-contained systems (which the user may want to copy and run on hardware external to the NSW) as well as to subroutines and components. 2c4b5

Batch scheduler. 2c4c

The purpose of the batch scheduler is to maintain a complete applications system in the NSW file system. The file should include any necessary job control information as well as the source and the object for component modules. 2c4c1

Some systems provide for run-time options. We would like the NSW to control the specialization of the standard system so that records can be kept on which options are exercised. 2c4c2

Applications systems usually require mass storage files and/or tapes. Since many vendor supported systems do not provide for cataloging, sequencing of new versions, etc., it seems desirable for the NSW to offer these services. They will be provided by tool(s) which (if they are developed new for NSW) must be written in COBOL or FORTRAN for a widely available computer system(s). 2c4c3

Once the selection of run-time parameters and the control of production files is made internal to the NSW, there are numerous opportunities to provide enhanced capabilities at relatively low cost. An example is a time-sensitve scheduler. Criteria might be established for when the system should run (e. g., the first Tuesday of every month). The NSW scheduler would automatically identify necessary data files, run the appropriate job (on one set of target machines) and distribute the results. Time sensitive scheduling could be implemented by a MULTICS scheduler. NOTE: Many jobs scheduled this way will be run by having the operators move tapes from NSW to independent production machines. An analogy can be made with second generation batch scheduling , except the NSW has automated much, if not all, of the work of the production coordinators and operators. 2c4c4

Software conversion aids 2c4d

Computer hardware has almost doubled in effectiveness and performance per unit cost every three years for two decades. Software conversion costs have prevented operational organizations from taking the fullest advantage of these cost reductions. 2c4d1

The NSW environment will help this problem in three ways: 2c4d2

The existence of a large pool of competing service centers will make new hardware systems available soon

after they are announced, at least on a small scale. New systems development can take place on the new hardware. Where it is cost effective, old tools can be converted. There is likely to be a substantial overlap during which remaining tools can continue to be accessed on their old hardware. 2c4d2a

The NSW will include a collection of tools for converting production software to run on different (and normally newer) hardware. 2c4d2b

The NSW will include hardware emulators. In some cases, it will be possible to build software for projected hardware through emulation. That will make it possible to delay selection of the production hardware until 1-3 years into the development cycle. The resulting savings could be as much as 50% of hardware costs. This is of course a longer term objective. 2c4d2c

Management Considerations 2c5

Usage Audit Trail 2c5a

A record of when tools were used and by whom, and the operating system version they ran under, should be maintained. Users must be able to control who knows they use a tool and indicate whether they want to talk to other people about how it works. When the operating system on a Tool Bearing Host (TBH) is changed, all tools must be tested, or they must be moved to a lower maintenance category and identified as not being converted to the current operating system. When the user has a problem, he must automatically be given the option of submitting a trouble report. Access to this information is a significant issue which must be studied and a policy formulated. 2c5a1

Maintenance Categories 2c5b

Each tool must be assigned a maintenance category. The highest category will imply a level of support equal to or superior to that provided by any vendor today. Consultants must be available to assist tool users. The maximum time to respond to trouble reports and fix known bugs must be tightly constrained. Notification to users must come a year or more before a system's support is withdrawn. When a vendor stops supporting software, the source code must enter the public domain. Statistics

should be kept and published on user complaints and acknowledged bugs. Lower maintenance categories will be appropriate for tools which are useful but non-essential. For example, an experimental debugging tool could be used even if there was no guarantee of its continued existence. DoD users are likely to install tools developed in-house in less than the highest maintenance category in order to reduce their obligation to support outside organizations. 2c5b1

Cost Predictions and Recovery 2c5c

Enough data should be collected to predict what a given activity will cost. There should be one or several tools to help tool installers define the characteristics of their tools and provide the users with cost predictions. The Works Manager must collect the historical data which these tools require as input. 2c5c1

Efficient use of NSW facilities must be encouraged by charging and reimbursement policies which reflect the true economic value of the services used. 2c5c2

Implementaton Guidelines 2d

The NSW is being built as a loose confederation of tools, with no technical bounds on the number of tools or the number of users which can be supported. An initial system, oriented toward preparation, publication, and management of documentatio,n and the construction of COBOL programs for the B3500/B4700, is scheduled to begin operating during the Summer of 1975. The initial tools will include a text editor, a COBOL compiler, and a document publication facility. Once the concept is demonstrated, hardware will be added to support more users and tools will be added to support a wider variety of software activities. 2d1

Initially the NSW will be composed of large modules with few interactions identified. As such, it will operate in a largely conventional manner without much cooperation among the modules. Over time, the coordination and cooperation among the modules will be tightened through the replacement of modules and the incorporation of new ones that identify and report all their important activity to the Works Manager. 2d2

Participating Organizations 2e

The NSW Steering Committee is currently composed of

representatives of the following government agencies which are supporting the project: 2e1

Air Force Data Automation Agency(AFDAA) 2e1a

Air Force Data Services Center 2e1a1

Air Force Data Systems Design Center 2e1a2

Rome Air Development Center 2e1b

Defense Advanced Research Projects Agency(ARPA) 2e1c

The following Commercial organizations and universities are under contract to assist in the development of various portions of the system: 2e2

Applied Data Research(ADR) 2e2a

Massachusetts Computer Associates (COMPASS) 2e2b

Stanford Research Institute(SRI) 2e2c

Bolt Beranek and Newman(BBN) 2e2d

MIT Project MAC 2e2e

UCLA Campus Computing Network(CCN) 2e2f

Speech Communications Research Laboratory(SCRL) 2e2g

Computer Corporation of America(CCA) 2e2h

Science Applications, Inc.(SAI) 2e2i

The NSW Advisory Committee is composed of representatives of the following Government agencies interested in the project: 2e3

Army Material Command 2e3a

Headquarters, A.F. Systems Command 2e3b

*****need a list***** 2e3c

PART II: PERSPECTIVE OF PARTICIPATING DOD ORGANIZATIONS' VIEW OF THE NSW 3

Air Force Data Services Center 3a

The AFDSC Environment 3a1

AFDSC is an operational element of the Air Force Data Automation Agency (AFDAA). Its mission is to plan, design, develop, and implement computer based management information systems, and to provide automatic data processing, computing and management science services to the Headquarters Air Force and the Office of the Secretary of Defense in the Pentagon, and other agencies as assigned. 3a1a

AFDSC operates three dual processor Honeywell G-635s and one (newer) H6060 in the Pentagon, all with the GCOS operating system. It also operates a dual processor Honeywell H-6180 with the Multics operating system and an IBM 360/75 running OS/MVT. One of the G-635s is unclassified; the rest of the systems are classified. The IBM 360/75 is used strictly for batch processing. Multics, a large interactive system which is well-suited to the manipulation of on-line data bases, is used for high priority operations research and budgeting models. Three of the four GCOS systems provide time-sharing partitions. GCOS time-sharing programs are typically small, and frequently supplement large batch systems (e.g. - prepare input transactions or scan output). GCOS Time-Sharing (TSS) is ill-suited to the debugging of batch programs because of core restrictions (24k words is a practical upper bound) and because the TSS monitor calls are different from the batch monitor calls. Many important batch subroutines (e.g. IDS file update) will not run under TSS, and in any case, correct TSS execution does not guarantee a program will run in batch. 3a1b

The first G-635 was installed in May 69. The conversion effort from the old IBM 7094s involved 11 months of parallel operation, part of it using a commercial service bureau and part of it with the new equipment installed. The development effort has now stabilized, with a majority of the Center's resources devoted to production and maintenance. The IBM 360/75 was installed in August 1972. It was acquired to support an existing workload, so there was almost no conversion and development cycle. Multics was acquired during FY74, and a majority of the work on the machine can still best be classified as development. 3a1c

About 20% of the AFDSC manpower (19.9% in FY73, 23.1% in

FY74, and 19.7% for the first five months of FY75) is devoted to developing new systems or making major modifications to old ones. Of the manpower devoted to development, 15.8% in FY74 and 21.4% in FY75 were for new Multics applications. Those amount to 3.6% and 4.2% respectively of total manhours. Except for Multics, a miniscule amount of machine resources were devoted to development. Considering only GCOS and the 360/75, the numbers are 7.5% in FY73, 7.7% in FY74, and 4.8% thus far in FY75. These facts are summarized with the statistics in Table 1. 3a1d

TABLE 1: AFDSC Resources Devoted to Software Development 3a1d1

	FY73	FY74	FY75	3a1d1a
Non-Multics Development vs Total Manhrs	19.9%	19.5%	15.5%	3a1e
Multics Dev Total Manhrs	0	3.6%	4.2%	3a1f
All Dev vs Total Manhrs	19.9%	23.1%	19.7%	3a1g
				3a1h
Computer Time For Dev vs All Computer Time (360/75 and GCOS only)	7.5%	7.7%	4.8%	3a1i

It is difficult to isolate the resources devoted to production and maintenance. The problem is to separate the execution of software which has been thoroughly tested and remains static for long periods of time from the making and testing of minor modifications. One indication is the abort rate for production jobs, which averaged 21% for FY 74. Obviously there is more to running production than simply reexecuting a standard program against different data. A related problem is the cost of these production aborts. The average cost of each production abort on the G-635s was $20.00 on System A and $26.00 on System B during FY 74. Because the average waiting time in the system queue for batch jobs on systems A, B, and C, for example, is 1.1 hours for the last 6 months of FY 74, and is usually much worse than average during the crucial budget update cycles, programmers faced with overnight deadlines must frequently make minor changes and run production against large data files without having tested their changes. In the present environment, there is no alternative. 3a1j

The last aspect of the AFDSC environment which impacts on Center use of the NSW is the hardware acquisition and replacement schedule. The schedule is important because major development, modification, and conversion efforts at AFDSC correspond to acquisition times of new hardware. The GCOS machines are currently being enhanced. It is anticipated that this will extend their useful life through FY 79. The IBM 360/75 is programmed to be upgraded or replaced during FY 77. Multics w******something is missing here:::::FY 76 and FY 77. 3a1k

AFDSC Operated Computer Service Bureaus 3a2

In addition to providing complete management information systems, systems analysis, and computer support to Hdq USAF and OSD, AFDSC provides computer time to AF and DoD organizations which have their own analysts and programmers. A component organization is the San Antonio Data Services Center (SADSC) in San Antonio, Texas. It offers remote access from several locations throughout the Southwestern US to an IBM 360/65 running OS/MVT. SADSC will soon be offering Burroughs B4700 service as well. A new Washington Area Data Services Center (WADSC) is under consideration. It would initially operate either a B3500 or B4700. SADC is operated as a fee for service activity, with usage-based charging and ADP cost recovery. It is planned that WADSC will also be operated on that basis. Since a goal of the NSW is to establish an economic marketplace for software

tools and computer time, the NSW's proposed approach to resource allocation is in complete consonance with AFDSC operating policies for its regional service centers. An important aspect of the NSW from the AFDSC point of view must be the possibility of providing NSW tools and services to other AF and DoD organizations. 3a2a

AFDSC Requirements For NSW 3a3

Usage Scenarios 3a3a

Table 1 classifies AFDSC development workload, and for each category identifies the most significant problems being addressed by the NSW. Tools which are required to support each category of work are also identified. Among the AFDSC requirements which are not included in the table are backup for production systems and communications among different vendors' mainframes. NSW technology is relevant to these requirements. However, the NSW's present development efforts assume that the production machines will not be connected to the network, and hence will not be under the control of the NSW. 3a3a1

Tool Requirements, by Target Machine 3a3b

Honeywell G635 or H6000 with GCOS 3a3b1

The primary requirement is for testing batch jobs . Source level interactive debugging is required for COBOL and FORTRAN. 3a3b1a

The number of interactive programs which merely supplement batch systems indicates that transaction processing may be more efficient in the Center's environment than time-sharing. Heretofore, TSS has been used because there is no reasonable debugging environment for transaction systems. If such an environment could be established, then much of the Center's interactive workload might shift to transaction processing. That would eliminate the problems associated with batch and time-sharing incompatibilities which have plagued AFDSC programmers. 3a3b1b

Tools are required for designing, implementing, and restructuring random, ISP, and IDS data bases. 3a3b1c

The new WWMCCS data management system (WWDMS) is

purported to allow non-programmers to maintain a computer database and retrieve reports from that database. WWDMS is enough like a programming system that it will be desirable to bring WWDMS under the control of NSW project management and performance evaluation tools. That implies installing WWDMS as an NSW tool (or more likely, a set of NSW tools). 3a3b1d

A standard GCOS machine is needed for final testing of batch systems. In cases where the data is unclassified and non-sensitive, that machine may also be used for production work. 3a3b1e

IBM 360/370 With OS 3a3b2

There are requirements for FORTRAN, COBOL, and possibly PL/1 development on the 360/370. For each language, source level interactive debugging is required. 3a3b2a

A number of vendors offer libraries of database management subroutines for the IBM 360 which can be used as building blocks for applications systems. These routines are similar in concept to ISP or IDS for Honeywell GCOS machines. It is important to include such tools in the NSW. 3a3b2b

Several vendors also offer self-contained Data Management Systems for the IBM 360. The Informatics MARK IV system is supported at both AFDSC and SADSC. As with WWDMS, it will be desirable to install MARK IV as an NSW tool, and bring the MARK IV development workload under the control of the project management and performance evaluation tools. 3a3b2c

There are known requirements to development and maintain transaction systems for the IBM 360. An example is the query system for military pay records at the Air Force Accounting and Finance Center (AFAFC) in Denver, Colorado. To effectively debug such systems, which are tightly integrated with the telecommunications handler, a virtual machine environment is required so that operating system level code can be debugged. 3a3b2d

A standard OS system is needed for final testing. In cases where the data is unclassified and

non-sensitive, that machine may also be used for production work. 3a3b2e

There are several tools which execute on an IBM 360 or 370 which are needed to support the development of software for other machines. Examples are verification and testing tools (including test data generators) and programs for restructuring COBOL and/or FORTRAN into reasonable structured variants of the ANSI standards. A hypothesis of the NSW is that these tools will be able to continue to run on the 360/370 IBM, but be used to build software for, as an example, a Honeywell 6000 with GCOS. 3a3b2f

Multics 3a3b3

An unclassified Multics is required to support development of software for other machines, and so that NSW project management and evaluation tools can be applied to the development of Multics software. Multics seems to be the correct environment for a sophisticated debugging system for GCOS software. Janus and RDMS are database management tools for Multics. 3a3b3a

Requirements For Tools For Controlling And Maintaining Existing Software Systems. 3a3c

A significant portion of AFDSC manpower is devoted to production systems: maintaining them, modifying them, and scheduling their execution. For the NSW to help in the short run, ways are needed to move these production systems into the NSW environment so that superior tools can be used by maintenance programmers. Some of the means to this end follow: Source, object, and job control must be copied into the NSW file system. The attributes of input and output files must be defined. Linkages must be established between component subroutines on the installation library and the production software. Narrative documentation is to be put on-line for easy maintenance, and to be linked to the source and object code for tighter management control. Other goals would be to produce machine readable abstracts and other documentation, to implement release and change control, to restructure badly structured programs, and to establish libraries of standard test data and/or results. 3a3c1

Air Force Data Systems Design Center 3b

THE AFDSDC ENVIRONMENT 3b1

AFDSDC Mission 3b1a

The Mission of the Data Systems Design Center is to analyze, design, develop, test, implement and maintain standard automated data systems. Standard systems are those that are common to more than one command. 3b1a1

The only exception to this mission assignment is the personnel function which is the responsibility of the Military Personnel Center. 3b1a1a

Excluded from the Center mission are the specialized systems involving intelligence, research and development, and those command unique systems which are designed and maintained by the command themselves to accommodate unique mission requirements. 3b1a1b

The Center is charged with exploiting opportunities for integration and for improving interface across the board. Integration is defined as the controlled development and organization of data systems so that separate systems use common data records and information, thus enhancing the efficient use of computer resources and avoiding duplicate processing of data. In the area of interface, AFDSDC wants to assure that output from one system is usable as input to other systems. The Center is also responsible for developing and maintaining non-functional utility software and the technical standards for the standard computers employed by the Air Force. 3b1a2

Since December 1971, the Center has been the designated ADPS manager for standard Air Force computers -- UNIVAC 1050-II; Burroughs B3500, B4700, and B263 base level machines; and the Honeywell H800/200 machine employed at the MAJCOM level. In 1974, this responsibility was extended to include the MAJCOM ADP Program operating Honeywell 6000 series machines (replacing the H800/200). The basic thrust of ADPS managership is the evaluation and coordination of impending software/hardware changes with the Commands in order to exercise configuration management. Impending software changes are assessed through the use of the Workload Anlysis Model and specific hardware impact determined through our Configuraton Analysis and Projection Section.

This information is projected nine quarters in advance and disseminated to the MAJCOMs for planning and budgeting purposes. 3b1a3

Organization 3b1b

The Data Systems Design Center is composed of two types of directorates, those directly associated with a specific functional area, and those which provide general support to the mission of the Center. 3b1b1

To insure responsiveness to functional customers, and to facilitate identity of efforts and close working relationships with Air Staff functional managers, the functonal directorates were created as a mirror to the Air Staff. The Directorate and Divisions in the Center correspond to the Deputy Chiefs of Staff and Directorates on the Air Staff. Each Center Director is, in fact, a functional expert. Included within his activity are the required automation experts. Responsiveness to the functional customer in the field is paramount, but responsiveness to the Air Staff functional manager also is essential because the design of a new data system invariably impinges profoundly on policy. To insure retention of current policy during system design, the designers must work hand in glove with their Air Staff counterparts. 3b1b1a

The support directorates are those of ADPS Management, Systems Control, and Systems Development. The Directorate of Systems Development supports retrieval systems and data communications control systems, the WWMCCS and MAJCOM Update programs, and Data Management systems. The Directorate of Systems Control operates the Center computers, handles quality control and release control, maintains a 24-hr-a-day Field Assistance Center, and supports the non-functional software (such as the operating system and utility programs). The Directorate of ADPS Management, in addition to those functions already discussed, maintains USAF/DOD standard Data Elements and supports the AFDSDC Technical Library. 3b1b1b

A relatively large Auditor staff is co-located with the Center to assure adequacy of audit trails and management during system development. 3b1b2

A Communications detachment from the Air Force

Communications Service is also co-located with the Center to provide ongoing communications expertise. 3b1b3

The Center has a special relationship with the Military Personnel Center. MPC is responsible for the development of standard personnel systems, but since the Personnel systems at base and MAJCOM levels operate on standard computers with all the other standard systems, our two programs must be closely coordinated. While MPC has complete responsibility for its functional systems, the Design Center is responsible for developing and maintaining the standards, for insuring Master Control Program and functional system interfaces, and for exploiting opportunities for functional integration and effective interface. After MPC has accomplished its own system testing and debugging, their products are sent to the Center for final testing and release to the field along with the other systems releases. 3b1b4

Responsibilities, Resources, and Workload 3b1c

The Center currently supports approximately 200 automated systems on 350 computers at 130 bases and sites worldwide. It is also responsible for 163 Air Force Manuals. The Center has seven major computers (an H6060, a B4700, two B3500s, two U1050s, and a B263) as well as numerous minicomputers including two FourPhase systems, an H700, a Nova 800, three RJET systems, an IMP and a PDP-11 ARPANET Access System. Manpower currently assigned is approximately 235 officers, 660 enlisted personnel, and 435 civilians . 3b1c1

The Design Center receives its functional policy guidance from the functional deputates and special staffs, and its overall automation policy guidance from the Director of Data Automation, who is also the commander of the Data Automation Agency. 3b1c2

The sources of Center workload, however, are many and varied. Obviously there is a continuing system maintenance workload associated with field reports of system deficiencies, official suggestions, etc. System modifications can be triggered by new laws, changes to OSD/Air Staff policy, or as a result of GAO/Auditor/IG recommendations. 3b1c3

While many of these requirements are levied upon the Center via letters, all major tasks take the form of a

Data Automation Requirement (DAR) and subsequently a Data Project Directive (DPD). Many are preceeded by requests for economic or feasibility studies, or other types of detailed analysis. 3b1c4

All major tasking documents flow through the Directorate of ADPS Management where they are reviewed and passed to the appropriate directorate(s) for more detailed analysis. The specific workload and resource impact upon all directorates is brought before the Requirements Review Board for consolidation and further evaluation. The Board's recommendations are passed to the Commander for decision. If approved, the task is entered into the Center Program and Resource Management System through which development progress is continually monitored. Projects are scheduled for formalized System Status Reviews and System Design Reviews at key stages of development. 3b1c5

IDENTIFIED APPLICATIONS

There are currently identified four major areas of application of the NSW system to the AFDSDC mission: Software development, testing, and maintenance; Documentation entry, editing, update, publication, and control; AFDSDC Office Automation including intra- and inter- organization communications; and Miscellaneous ARPANET Usage. 3b2

B3500/4700 Software Production (Subproject A)

The NSW will be used to assist and control all phases of the system production process, from initial design to continuing maintenance. Programs will be interactively written, debugged and tested on-line. An integrated database on each system, including source and object code for each program, design documents, system and user manuals, DIREPS, and pointers to other systems sharing (production) files will be available on-line. 3b2a

Documentation (Subproject B)

The NSW will be used to publish and maintain all documentation distributed by AFDSDC. This will include FUSM and System manuals, DPPs, Reports, etc. These documents will be entered via an off-line cassette tape, fed into the NSW file system, edited, and published in Microfiche and hardcopy formats. Updated versions will be prepared using the NSW text editors, greatly reducing work. Flow charts, diagrams, and similiar line drawings will be interactively generated at advanced CRT stations, and stored on-line as part of the document file. The on-line documents will form a

rapidly available library, accessable by AFDAA and other AF and DoD users. 3b2b

Office Automation (Subproject C)

The NSW will also provide an Office Automation system. Correspondence within AFDSDC, and between AFDSDC, AFDAA, and AFDSC will be prepared, coordiated, distributed, and filed using NSW tools. (Correspondence with other organizations will also be prepared and coordinated internally using the system, but will be then printed on a high-quality printer before being sent to the external organization). 3b2c

Miscellaneous ARPANET Usage (Subproject D)

AFDSDC organizations will make use of facilities of the ARPA Network not available as NSW tools via a "TELNET-like" tool provided by NSW. Used in this mode, NSW will only provide a general Network access facility, a pseudo-TIP. User organizations will be required to negotiate individual accounts and payment procedures with the owners of the software and hardware to be used, exactly like today. 3b2d

SCENARIOS OF OPERATION: CURRENT AND WITH NSW 3b3

B3500/B4700 Software Production 3b3a

CURRENT ENVIRONMENT 3b3a1

Manual Methods

The first step of the current method of running a job is to submit a hand-written coding sheet to be keypunched. The program is keypunched and returned to the user. The user checks the deck against his coding sheet, and repunches any cards as necessary. When the deck is correct, the user prepares a workorder, and jobcards. He then places the deck and workorder at a designated pickup point within his building. Twice daily, a courier comes and delivers the job to the computer center. At the computer center, the jobs are logged in, sorted for priority, and put on carts for the machine operators. The machine operators load the decks into the machine, then return the decks to production control. Periodically, printouts are also returned to production control. Production control mates the job with its output, logs the job out, and puts it into outgoing distribution. The next time the courier makes his rounds, he delivers the job back to the users' building, where the user picks it up. The user checks the output, repunches new cards and resubmits the job until it is correct. 3b3a1a

Online Remote Compile And Test System (ORCS)

Under the new ORCS system, the user still submits a handwritten coding sheet to be keypunched, but the deck produced will then be loaded directly into the machine, and only a listing returned to the user. The user checks the listing, and makes up change cards to correct any errors. These change cards are submitted via the ORCS RJE terminal. The change cards are merged with his program file, the job run, and the resulting output diverted to the RJE terminal. The user may then review the results, make up additional correction cards, and repeat the process. 3b3a1b

NSW ENVIRONMENT 3b3a2

NSW Job Submission- Simple Case.

Under NSW, the user has two options for entering his program. The first option corresponds somewhat to the current procedures, where the user will hand-write a coding form and send it to be keyed. Instead of being keypunched to card, however, it will be keyed to (off-line) cassette, then entered into the NSW file system automatically, while the key operator is recording another program on the second cassette unit at her station. The user also has the option of entering the program on his own offline or online CRT/cassette unit if he wishes. (The semi-skilled typist can typically type as fast or faster than he can print by hand, so this option is reasonable and could reduce keystroke workload considerably.) Once entered into the NSW files, the program can be reviewed and corrected using a text editor. Then a job can be created. This job enters the batch queue, is executed, and the results returned to NSW. The user can then use the text editor to review the results, and correct his job, before resubmitting it for another run. 3b3a2a

Note that programs, just like any other document, letter, or group of TEXT, will be keyed by the organization secretary, not by a special keypunch section. Secretaries tend to be both faster and more accurate than keypunchers, and by gradually doing away with keypunch positions, AFDSDC will be able to save manpower, money (for keypunch machines and cards), time (due to the faster turnaround of having the programs punched within the organization), and errors (card images on a CRT are easier to type, review, and correct than on physical cards). 3b3a2a1

The process of creating a batch job (assuming all source and input files are already in being) will involve invoking an NSW batch submission tool, telling it what to run, where, and which files to use. This tool will then handle all file movement, login, entering the job in the batch jobstream, and retrieving/disposing of output files. 3b3a2a2

NSW Job String Submission

The simple case discussed above will provide some

increase in productivity over ORCS (via a better way to view, update, and reenter a job) and significant increases over the manual method (because of the considerably faster turnaround.) It will also allow better management controls by improving reporting, automating standardization checks, imposing management policies, etc. However, the largest payoffs will not come from the single job but from the submission of a job string. In this case, the user can generate a string of sequentially running batch tools, perhaps even running on different machines. 3b3a2b

For example, the user might be writing in structured COBOL. To test execute his program, he might generate a jobstream;

PREPROCESSOR;TEST-DATA-GENERATOR;PERCENT-EXECUTE;
B4700-COBOL-COMPILE;EXECUTE;LISTING-REFORMATTER
; each job of which is dependent on successful completion of the previous one. A failure at any step will abort the stream. (The fact that each tool may run on a different physical machine will be transparent to the user). 3b3a2b1

Thus the user would have software development tools available to him to automate such things as translating from a structured version of a language (which is easier to read, write and debug) to the standard version, to perform the time consuming job of instrumenting the program to assure all control paths are used, of generating appropriate test data, and of reformatting the listing so it is easier to read. The system would have thus taken over much of the manual work normally necessary, allowing the programmer to concentrate on programming. This should yield a significant productivity increase. 3b3a2b2

Interactive Writing/Reading/Correcting

Corrections to the program file will be made on-line through the use of a text editor which "understands" the language the program is written in. This system will check changes to the program (as they are entered) for syntax, provide recognition of reserved words and datanames, and maintain the structure of the program in an easily understandable form. It will also enforce certain programming conventions such as limiting blocksizes, requiring

explanatory comments, etc. After a program has compiled and run, the compiler generated listing will be structured for easy readability, and any errors in syntax which slipped past the editor will be flagged. A listing of the machine instructions generated by the compile will also be available within the structure to allow the user to find the exact machine code generated by any questionable statement. 3b3a2c

Interactive Debugging Package

A system will eventually be available to allow a programmer to interactively control the execution of his program; to manually step through areas of code which are malfunctioning, to change variables, to trace branching, etc. Thus, the on-line programmer will have all the capabilities to quickly isolate bugs, just as if he had the machine all to himself. Several such copies of this package may run simultaneously, so in effect it will be possible to give several users "dedicated time" simultaneously, significantly improving utilization of the hardware resources available. With this facility available, it is expected that the only work requiring a dedicated machine will be operating system (MCP) maintainance, Data Communications Control System program developement, and environmental system tests. 3b3a2d

Program Testing

Tools to trace the execution of the program, and to generate test data to assure the system has been completely exercised will be available to both the programmer and the quality control branch. Through the use of such tools, software will be more completely tested and therefore more reliable when released to the field. 3b3a2e

Project Documentation and Management

All concept documents, specifications, design documents, progress reports, etc, pertaining to the project will be prepared online and stored in a Project Documentation file. This file will be available to the project managers (to allow better tracking of how the project is progressing) and to the functional analysts and programmers (to allow rapid access to design/conceptual documentation). The manager will also have available an interface to the PARMIS II system to allow him to enquire about manpower and scheduling aspects of the project. 3b3a2f

Top Down Design

In the long range, the Center may mov eto Top Down design techniques using structured programming concepts. Programs could be defined using a high level block-structured Program Description Language. Source code would then be added to do the functions described by each PDL statement (which would in turn become a comment.) A tool would be implemented to extract the PDL description into a textual description, and into a flowchart at a user-specified level of detail. Both of these facilities promise to significantly simplify, standardize, and speed system documentation, as well as improving the product and allowing new personnel to learn new systems much more quickly. 3b3a2g

Minicomputers

As the USAF moves to using more minicomputers, emulators and/or slaved development minis will be used to evaluate hardware/operating system, and to write, test, and debug software for such systems. Microprogramable computers will make such processes faster and more cost effective than current higher level emulators. Another possibility which in some cases may be implemented is to slave a Mini to a larger machine in such a way that the master computer controls the mini, acting as a "pseudo-operator". In either case, it will thus be possible to bring the full powers on NSW online program development tools to bear on writing software for minicomputers. 3b3a2h

Documentation 3b3b

CURRENT ENVIRONMENT 3b3b1

Initial Typing And Publication

The current environment is a manual, typewriter based system with photo-offset printed products. Documents are hand written by the author. The documents is then typed by a secretary and returned to the author who marks corrections, to be made by the secretary. This process is repeated until the document is (in the author's view) correct. The document then is manually distributed for coordination/approval. At any step in this process, additional corrections (ranging from minor wording changes to inserting/deleting/ moving several lines or even pages) can occur. Each such change requires manual

correction ranging up to complete retype of large portions of the document. When the document is finally ready to be released, it is retyped on a special form, photographed, and printed in hard copy. 3b3b1a

Update And Republication

As changes are made to a document, two possible procedures can be followed. If the changes are extensive, the document must be republished, requiring complete retyped as if from scratch. This can also occur if there have been several separate smaller changes made to the document since it was last republished. (In this case the document has become a mass of changes to changes, and becomes difficult to read and update in the field.) 3b3b1b

Changes

If the changes made to a document are not so extensive as to require republication, only a list of changes are published. This list of changes directs the field users to make pen-and-ink corrections to his copy of the manual if a change involves only a sentence or two. If the changed area is large enough to make a pen-and-ink correction impossible (say adding a paragraph), a new replacement page is provided with the list of changes and the field user is directed to substitute this page for the corrected one. Obviously this procedure takes many man-hours Air Force wide, and is prone to error. 3b3b1c

Manual Microfiche

Because of the rising cost of paper and postage, and to alleviate the error problem, early in 1975 AFDSDC will start publishing selected manuals in Microfiche format. The procedures will be essentilly the same as before except documents will be photographed and the film used to prepare Microfiche instead of being printed on hardcopy. Each time a manual is changed, a new fiche will be prepared, so that the process of correcting a manual in the field will be simply to replace the old fiche with a new fiche. 3b3b1d

NSW ENVIRONMENT 3b3b2

Initial Entry

Initial entry of a document into the NSW will be via an off-line CRT and cassette tape. A secretary

will type a screenfull of text (~25 lines), review and correct obvious errors, then store it on cassette tape. This process will repeat until the complete document (or section of a document) is on cassette. It will then be read into the NSW file system. 3b3b2a

Alternatively, the author may enter his text directly on his own CRT/cassette unit, instead of hand writing it. This would be more efficient if he were a fairly good typist, but should not be expected to be the normal case. In such an instance, the entered document will probably be referred to a secretary to "clean it up" (correct spelling errors, typos, etc.) 3b3b2a1

Editing

Once entered and generally corrected, the document will be referred back to the author for review and correction. If the author is reasonably skilled in using an NSW Text Editor facility, this will be done on-line at the author's CRT. If not, the secretary will have the system generate a hardcopy of the draft, on which the author will mark corrections. Those corrections will then be made by the secretary. (Note again that it is nearly as quick for an author to use the editor to actually make the corrections himself as to mark them on hardcopy for a secretary to do- and response is instantaneous.) The corrected draft will then be restored into an on-line file. 3b3b2b

Diagrams and Figures

Diagrams, figures, flow charts, and other such line drawings will be manually sketched by the author. The sketches will be sent, together with a file reference, to a Graphics Specialist, who will use the NSW (NLS) Graphics facility to generate an on-line version of the diagram. (This person will use the same terminal as the Publication Specialist, and in fact, will probably be the same person.) Once the diagram is on line, the Graphics Specialist will generate a hard copy and return it to the author. The author will review the hardcopy, and mark any corrections necessary, then return it to the Graphics Specialist, who will make the corrections, and generate another hard copy of the corrected figure, and so on. When the diagram is finally correct, it will be merged into the text of the document at the appropriate place. 3b3b2c

Photographs

Pictures- as opposed to line drawings- will be photographed, catalogued, and flash overlayed onto the COM Master. The author will be required to specify the size, location, and catalogue number of a desired illustration's negative. The formatting system will automatically leave an appropriate space in the output text file and generate a COM machine control instruction to cause insertion of the photograph at the appropriate point. This process will not involve the Graphics Specialist, and will be held to a minimum, as it involves some extra expense, and slower response. However, it will be available if required. 3b3b2c1

Hardcopy Documents

In the long range, this system is primarily designed to generate COM documents. However, it will also be used to manage, update, and coordinate documents which must currently remain in hardcopy because of requirements of the user. In such cases, the previously discussed process will apply, except that instead of working with the entire document, only the changed portions will be extracted to a temporary file. Corrections will be made and hardcopy masters of the changed pages will be produced on a high quality printer for publication via conventional means. 3b3b2d

Coordination

The draft document (or changes) will then be circulated via the NSW/Journal mail facility for coordination/review/ correction. If the reviewer has comments or corrections he would like made, he will generate a list of comments and link them to the document, to allow their easy insertion in the document. This process may proceed either serially or in parallel. In other words, during the early stage of writing a document, the author may want to distribute it to several people simultaneously for comments and suggestions. On the other hand, once a final draft is ready, it may need to be serially passed up an approval chain. 3b3b2e

Formatting And Proofing

Once a document has been approved, it will be sent (via a file reference in NSW Mail) to the Document Preparation Section. Here, a trained Publications Specialist will append any special formatting commands

required, and then generate a "COM Proof", a hardcopy approximation of how the document will appear when published via the Computer Output Microfilm facility. This proof will be returned the author for final checking. When he approves it, the proof will be returned to the DPS and a COM master will be generated. This will also be checked for quality control, and if acceptable, the master will be sent to the reproduction section. 3b3b2f

COM Master Generation

In the early phases of the project, generation of the COM masters will be done by a commercial service bureau (due to the cost factors involved). As the use of NSW grows and becomes more widespread, an NSW COM publication center will be set up, possibly at AFDSDC due to response time considerations. This facility will provide service to all NSW users on a cost-reimbursable basis. 3b3b2f1

Reproduction And Distribution

The Master fiche will be photographically reproduced (using the same equipment used to reproduce manually prepared fiche). It will then be distributed either by mail, or with the Block Release, as appropriate. 3b3b2g

Storage And Republication

Once published, the on-line copy of the document will be retained on low cost storage (either a tape library, diskpack library, or on an online, Datacomputer-like service facility). Once captured, it will never again be necessary to retype the document. Changes, no matter how large or small, will be made by simply repeating the preceeding process starting with the EDITING stage. 3b3b2h

Office Automation 3b3c

CURRENT ENVIRONMENT

The current environment is almost totally manual and paper-based, using typewriters, manual coordination, manual mailing/distribution, and manual filing. Some users are currently evaluating automatic typewriters such as the IBM Magnetic Tape/Card Selectric Typewriter systems, but there are only a very few such machines

currently used for production, and NSW should eventually replace these with more powerful CRT systems. 3b3c1

NSW ENVIRONMENT

Under NSW, tools will be available to support the preparation, coordination, and distribution of correspondence, reports, and the like within the Center, and to some extent between the Center and other organizations with access to NSW, notably AFDSC, AFDAA, and AFSC. Used this way, there will be little difference from the mode of operation described in the DOCUMENTATION section. The primary difference will be that coordination and distribution of shorter documents will predominate over publication of larger ones. 3b3c2

Document Preparation

A document- a letter, memo, report, etc.- will be hand written by the author (or entered directly as above), loaded to a cassette by a secretary, then spooled in. As in the preparation of a manual or larger document already discussed, there may be one or more correction steps until the document is correct and ready to distribute. It will be then passed up the approval chain and distributed (by NSW Mail for AFDSDC or other online users, and by hardcopy US Mail for non-users) to the addressee(s), and a copy kept in archival storage. 3b3c2a

Automatic Typewriters-Plus

For most shorter documents produced in the Office environment, the CRT/Cassette unit will be used offline as an automatic typewriter, but with the capability to connect to NSW to do more more extensive editing functions than those available on the terminal itself, or to enter the document into the online distribution environment. This will reduce the contention for relatively scarce and expensive external computer resources, and will simplify training of secretarial personnel, in that only a portion of them will actually use the more powerful but more complex online editors. 3b3c2a1

Secretaries' Editor

A recognized problem with the initial NSW editor (NLS) is that, because of the power it offers the user, it tends to be more complex to use than other editors currently available. Since the average secretary will not actually use the full

power of NLS, it is desirable to define a simpler command language for such non-technical personnel. This language should allow only a basic subset of the complete NLS command language. AF/DAX will provide significant inputs to the design of this language, as will AFDSDC and AFDSC non-technical personnel who have used the normal NLS system. 3b3c2a2

Document Reception

The process of receiving a document (for an on-line user) will be simply to receive a pointer (LINK) to the single on-line copy of the document, and to use this pointer to retrieve the necesary information. Such LINKs will also be used for filing, referencing the document in future letters, etc. To make such a reference, the user will simply insert a Link to the original (or referenced) document, within the new (referencing) document. 3b3c2b

Time Managers

There will also be available a time management service to remind users of appointments, project milestones which are (or soon will be) due, etc. This facility will also be used to set up meeting schedules to best fit the available time of those involved, and be used to control suspenses on work. 3b3c2c

General ARPANET Usage 3b3d

CURRENT ENVIRONMENT

Current access to ARPANETed facilities is via the IIP at Patrick AFB, Fla. (RML). Accounts, information about services, payment procedures, funding sources, and the like must be handled individually between each using organization and serving site. The Network access facility offers access control, a little usage information, and very primitive user services. 3b3d1

NSW ENVIRONMENT 3b3d2

Accounts, funding, etc.

NSW will not handle accounting for Non-NSW tools and facilities on a production basis. It may in certain instances handle setting up accounts for trial or experimental usage of systems not available within NSW, but which appear to be useful additions. However, in the general case, the process of setting up such

"outside" accounts on machines which are not TBHs will be the same as in the current environment. 3b3d2a

Access Facility

Access to "Non-NSW" portions of the Net will be via a "TELNET-like" tool, (actually just an invocation of the PDP-11 ELF system with a 'transparent' grammer). Access control to the NET will be via the normal NSW logon; to the foreign host via that host's Logger function. This tool will offer a more understandable command language and some additional services over those currently available on a TIP, but it will essentially be a "TIP replacement". 3b3d2b

Long-Range Evolution

It is expected that such extra-NSW use will be relatively common in the early phases of the project, but less and less so as more machines are equipped with TBH software. As soon as a machine becomes a TBH, any non-proprietary software running on the machine can be used within the framework provided by NSW. Thus as NSW expands, more and more tools will be available through NSW, and only a few will be accessed, and payed for, outside of the NSW environment. 3b3d2c

Rome Air Development Center 3c

Introduction 3c1

RADC is engaged in a research program to develop tools which will aid in reducing the high cost of software development. It is well known that tools such as compilers, editers, and debuggers aid immeasurably in the production of software. Within the last few years, a number of tools have been developed at RADC. However, making these tools available to a larger community of users has been a continuing problem. RADC's involvement in the National Software Works (NSW) program stems directly from its desire to make the tools it develops readily available to the DoD organizations which build production software systems. Through closer interaction with the user community, it will be easier to produce tools to satisfy the real needs of software designers, and, conversely, the designers will be able to use the tools so developed and thereby justify the support and maintenance of the tools. 3c1a

Thus, the RADC software development effort will be divided into two complementary classes: development of software tools in response to user needs, and the development of NSW as a vehicle for making the tools available to a larger community. 3c1b

Five general areas of research have emerged which indicate the thrust of the RADC software research and development effort. 3c1c

1. MULTICS-NSW Development 3c2

MULTICS will be the prime research and development machine at RADC and most of the emphasis on new software products will be focused on this operating system. MULTICS offers a natural second choice to the TENEX operating system in its easily extendible time sharing environment. MULTICS is maintained and supported by a commercial vendor, Honeywell, who has a vested interest in its success. RADC will fully integrate the RADC MULTICS machine into the NSW environment. 3c2a

To provide service for an increasing NSW user community, it will be necessary to replicate the functional components of the NSW. It is felt that the major NSW components such as the Foreman, Front End, and Works Manager must not become specialized to the TENEX environment. 3c2b

RADC proposes to fund development of a MULTICS based Works Manager (WM), including research in problems related to multiple WM such as accounting, file handling, and user profile and logon information. Issues such as the assignment of a WM to a user, WM relationship to a user during a session, data base integrity, data base sharing, maintenance of catalogs of filed objects, and request validation must be addressed. Changes will be required within the various Front Ends to interact with different WM. 3c2c

A principal activity at RADC will be to adapt and develop software tools to run under MULTICS. Some existing tools will be moved from their GCOS environment to MULTICS and encapsulated. New tools will be purchased from venders or developed. The software to make MULTICS a tool bearing host (TBH) will be procured. RADC also plans to move the NLS editor to run on MULTICS. The anticipated number of users on the NSW will require multiple copies of NLS to prevent long delays occuring on an overloaded system. Development of this tool will require implementation of NSW design considerations such as WM selection among identical tools. 3c2d

The RADC H6180 is connected to the ARPANET through a hardware interface developed by MIT. The interface connects a host port on the RADC TIP to the multiplexor I/O controller on the H6180. Honeywell supplied the Network Control Program (NCP) and MIT supplied the remaining network software packages . 3c2e

2. Structured Programming Environment 3c3

A major contributer to the high cost of software is the lack of proper management control over the software design and implementation process. Because a team effort is often required for software projects, the need for good communication among programmers is essential. This includes communication on areas of responsibility, interface constraints, documentation and status. All too often a lack of communication results in expensive software retrofits. All too often the project managers have little idea of the progress of the individual members of the team. Because programming tends to be an art, every progammer has his own method of coding and documenting which often cannot be understood by another person who may have to maintain the software or take over the responsibility in mid-stream. It is to address these areas that the structured programming concept has been formulated. 3c3a

Much has been said pro and con about the advantages of structured programming and there is much misconception of its scope. In the context of this effort structured programming refers both to methods of programming, and to an environment which produces sufficient information to allow a manager to retain close control over his programmers. RADC plans to become involved in the creation of a structured programming environment (SPE) using the NSW as the communications vehicle which links together the distributed set of users. 3c3b

Creation of an SPE is a logical extension of the concept of the NSW. It consists of a set of tools to aid the programming process, a library environment where documentation is maintained on project status, and tools for the measurement of software production. 3c3c

RADC intends to determine whether this approach significantly improves the quality of produced software by developing a structured programming environment. This includes development of a program support library, collecting tools under the SPE which aid in producing structured software, and introducing management software aids for maintaining control over the entire software design process. It is felt that such a facility would provide an ideal mechanism for running initial experiments and later on a production basis. 3c3d

A scenario of SPE-NSW could occur as follows. A programmer logs into the NSW and builds his COBOL program under control of the RADC SPE. He would have access to structured programming design aids on the network. The programmer then would send the file to a preprocessor which would enforce desired programming practices . The output file is then sent to a compiler, with the object code sent to his computer for execution possibly under control of a debugging tool. As the programmer uses these design tools, editors, debuggers, and compilers on the NSW, other tools could be collecting management information. 3c3e

User interaction with a set of NSW structured programming tools would be under control of an RADC tool which would enforce structured programming practices throughout a software design project. The Program Support Library would maintain copies of the users source, object, and output files, and would maintain linkage information on developed software modules in order to enforce consistency of

interface specifications. These files would be NSW files and would not necessarily reside locally. 3c3f

This SPE tool would add an extra level of overhead between user, Works Manager, Front-End, and tool and would have to be carefully designed to minimize such overhead while retaining adequate control over the software development effort. 3c3g

RADC is currently becoming involved in a major structured programming experiment with an Air Force software develoment group at SAMTEC in Vandenberg, Cal. The purpose of this experiment is to test the effects of applying a defined set of software tools and design and programming rules on the software development cycle. A test and evaluation program will be conducted in parallel with software development to obtain data which will provide management with quantitative information on the value of modern programming practices, how they can be used to improve control and production of quality software, and what further improvements need to be made. 3c3h

This experiment will be conducted during FY75-76 and is not planned to use the resources of the NSW. All design tools will reside on the computer at SAMTEC, with data collected and analyzed at SAMTEC by personnel from RADC. A future experiment has been proposed to be conducted at the Data Systems Design Center in Montgomery, Ala. 3c3i

Future structured programming experiments can benefit heavily from the technology provided by the NSW. With an SPE at RADC available to users across the country through the NSW, duplication of resources is reduced. The personnel conducting the experiment can be more intimately connected to the actual software designers without regard to geographical distances. 3c3j

3. Language Control Center 3c4

No single tool is more important to a programmer than the programming language he uses. It is the programmer's vehicle for communicating with the computer and for expressing the problem under solution. In order to provide elegant and efficient solutions to problems, the language should be concise in its notation, understandable in its underlying structure, secure or error-resistant in its mechanism , and relatively simple or straightforward in its control constructs. Ideally, the language should be close

to the programmer's natural language since the source listing of the program may serve as the medium of communication among programmers or between a programmer and a maintenance man. The language should be easily learned, have high retentivity, and be stable in its specification - at least over relatively long periods of time. 3c4a

The compiler or language translator is the software package that embodies the language on the machine and is the tool that the programmer actually uses in processing his code. The compiler may rearrange or optimize the programmer's code, notify him of syntactic errors in coding, warn him of inefficient coding practices and perform other functions which are not necessarily related to the language translation process. Hence the compiler is a tool which not only makes the language available on a computer but also provides other programmer services as well. The programmer quite often associates the performance of the compiler with the capabilities of the language. Obviously, these are completely separable entities. 3c4b

The writing of compilers falls into the domain of a particular software specialist - the compiler writer. It is his function to translate the language specification into a software package which embodies within it the language specification. In addition he makes various decisions as to the services he will provide to the programmer via the compiler, even though these are usually not covered by the language specification. 3c4c

Compiler quality is often a highly variable characteristic. In terms of the language specification, the compiler writers product can only be as good as the language specification from which he works and hence the necessity for completeness, non-ambiguity, and understandability in the language specification. It is also true that compiler quality is often dependent on the writer. The high cost of this type of software is often due to inefficient implementations, incorrect implementations, high cost of training and retraining programmers, development of resistance in programmers to changing languages, and other problems. 3c4d

The usual solution to a problem of this kind is standardization. Standardization at the level of a language specification has been tried for years with little effect on reducing the proliferation of language dialects. There was no attempt (other than the next cycle of language revision)

to verify that the specification in itself was complete and non-ambiguous, no measures were taken to insure that all compilers for one language implemented exactly that language, and little if any standard tests were applied to each compiler in its own machine environment to insure closure of the language standard and its compiler realization. 3c4e

RADC believes that DoD needs a Language Control Center (LCC) for the DoD which would serve to stabilize the trend toward computer language proliferation and inefficient and costly compiler implementations. A center of computer language expertise coupled with the NSW communications mechanisms would be an ideal method of providing language controls. The essential elements of such a facility would include compiler generating tools, compiler validators, language specification writing tools, compiler statistics collectors, code optimizers, special purpose front ends such as structured programming preprocessors, and language and compiler documentation tools. 3c4f

Compiler Generators 3c4g

Many tools now exist for generating compilers for different machines. One tool, JOCIT, is currently available at RADC. JOCIT (JOVIAL Compiler Implementation Tool) is a tool for producing either JOVIAL J-3 compilers to execute on a target machine or for producing cross compilers which generate object code for the target machine. This tool would tend to stabilize the J-3 language because it can produce compilers at 30% of the cost of standard compilers very quickly, thus making it more attractive. Other compiler producing tools are under consideration. Changes in the language are handled easily these tools and the LCC can provide a means of testing changes before distribution to users. 3c4g1

Compiler Validators 3c4h

Compiler validators are programs which test a compiler to determine whether all language features have been correctly implemented according to the language specification. A validator consists of a series of test modules written in the language of the test compiler which are compiled by the test compiler and then executed. During both compilation and execution, descrepancies between the language features of the candidate compiler and the language specifications are

noted. RADC has validators for the JOVIAL J-3, JOVIAL J-73, COBOL, FORTRAN, and BASIC languages. 3c4h1

Language Statistics Collectors 3c4i

Several efforts are underway at RADC on developing automatic statistics collectors designed to show how programmers utilize the different computer programming languages and their specific features. SIMON is a software monitor at RADC which collects statistics on the use of JOVIAL J-3. Data collected include management data such as error, productivity, and accounting information. SIMON can accumulate statistics over many compiles. 3c4i1

Language Specification 3c4j

In order to write a good compiler it is essential to have an unambiguous, complete language specification. SEMANOL, a meta-language for precisely describing a higher order language, has been implemented at RADC. A JOVIAL J-3 SEMANOL specification has been made and tested for completeness in both syntax and semantics. Extensions of this concept to JOVIAL J-73, FORTRAN, and COBOL are being considered. 3c4j1

RADC has been involved in the language controls and development area for a number of years. The recurring problem of direct user support and interaction has heretofore inhibited a broader attack on the problem of language controls. The NSW will provide an impetus to research in this area through closer coupling between RADC and a user base. 3c4k

4. Document Production Research 3c5

The high cost of local document production has created an interest in the utilization of specialized sites which have highly developed documentation facilities. Document production involves the creation of source files, editing of files, local printing at reduced capability for error and format checking, and transferring of files to a documentation center for report production on various media such as microfiche or hard copy. Each step in the process of producing a document requires a means of connecting the user to the various centers. The NSW is ideally suited to providing a local site with access to these centers. 3c5a

In cooperation with the Air Force Directorate of Administration, RADC will be investigating the general area of computerized report production in light of the requirements of the DoD. Investigations will be made on the location of specialized centers where documentation aids exist, as well as the development of mechanisms for utilizing these facilities through the NSW. 3c5b

5. WWMCCS Research 3c6

The Worldwide Military Command and Control System (WWMCCS) was first envisioned in 1961 to become a confederation of individual systems that would support the needs of the services and the Unified and Specified Commands. Since 1971 the WWMCCS role has been redefined as a more closely knit system to support the National Command Authority. WWMCCS, a computer based network of command centers, links all military command and control systems to perform two essential functions: to provide warning and intelligence information necessary for the President to make decisions, and a capability to transmit those decisions to the military forces. The program includes the acquisition 35 new standard computer systems, the development of standard software to meet the common requirements of the WWMCCS commands and installations, and the evolutionary development of these systems into an integrated WWMCCS Automatic Data Processing (ADP) System. 3c6a

WWMCCS is administered by the WWMCCS Council which is composed of the Deputy Secretary of Defense, the Chairman of the Joint Chiefs of Staff, and the Assistant Secretary of Defense (Intelligence). The agency reponsible for the management of the WWMCCS ADP effort is the Joint Technical Support Activity (JTSA). RADC has been given informal direction to engage in research in the development of improvements and new concepts to WWMCCS ADP software and hardware. Five areas of research activity have emerged. 3c6b

Real Time Operating System Techniques 3c6b1

The current WWMCCS GCOS is not capable of meeting the required response time to certain classes of command and crisis levels. RADC is developing a transaction processing subexecutive which will allow multiple transaction driven applications to operate in priority order, share data, and more efficiently utilize limited core space. This module will be tested using sample programs from a selected Air Force user site.

If test results confirm predicted capabilities, the tested design will be transitioned to JTSA to be included as part of standard WWMCCS software. Further enhancements to priority processing and transaction processing are being planned. 3c6b1a

Software Reliability and Recovery 3c6b2

The requirements imposed on WWMCCS during a national crisis situation demand high reliability of the entire network of computers and command posts. System crashes can mean loss of data or lengthy reboot and data restore processes which can consume hours. No formal procedures or specific system software exists to handle the WWMCCS restore restart/recovery problem. RADC plans to develop software aids necessary for operators, administrators, and site engineers to ensure graceful degradation and recovery, on-line reconfiguration, and data restoration. This will require interaction with AF user sites which will provide data bases and applications for subsequent testing with RADC developed software and procedure. The test results and design recommendations and software will then be transitioned to JTSA for inclusion into WWMCCS. 3c6b2a

WWMCCS GCOS Design Improvements 3c6b3

A number of currently available software designs and approaches will be investigated to determine their applicability to GCOS performance improvement. One technique to be studied is the Virtual Machine Monitor (VMM) concept . As an aid to software development multiple operating systems may reside under the VMM, allowing each user to run his own system, optimized for his application. Modularization of GCOS will be studied in order to develop a system which will allow tailoring of GCOS to the requirements of various sites. Various software modeling techniques will also be explored. 3c6b3a

Front-end Processing Software 3c6b4

Providing access to computational power and data to a diverse set of user types is an important requirement of the AF WWMCCS environment. Remote batch, time-sharing, and remote interactive consoles must be interfaced to GCOS through front end processors such

as the Datanet 355 and the Network Processing Supervisor (NPS) software. RADC has an ongoing program to test NPS and front end configurations that are provided by Honeywell. Results are forwarded to JTSA for their evaluation. 3c6b4a

WWDMS Investigations 3c6b5

The World Wide Data Management System (WWDMS) under GCOS is an attempt to provide a generalized DMS under WWMCCS. It conceptually provides all of those functions necessary to prepare and handle data for normal daily processing. Certain shortcomings have been uncovered during extensive testing of WWDMS including limited file restructuring capabilities, off-line querying requirements, and slow turnaround time from query to report. RADC is involved in the testing aspect of WWDMS and plans to develop new packages to improve the performance, including command language enhancements, file restructuring techniques, and transaction oriented query generation and interpretation. 3c6b5a

The RADC involvment in the research aspects of WWMCCS development rely heavily upon close interaction between RADC and the WWMCCS community. The software developments must be accessable to the community for test and evaluation. A role for the NSW is clearly evident in providing a means of communication between the research center and the user commands. Extensive use can be made in the WWMCCS research environment of the capability to transfer files and to control process interaction. 3c6c

PART III: OPERATING PLAN 4

NSW Management 4a

At the 8 April 1975 NSW Steering Committee meeting a plan for managing the NSW for the next three years was accepted. 4a1

1. STEERING COMMITTEE: 4a2

The Steering Committee will consist of two representatives from each of the funding organizations. The chairman is currently from ARPA. The Steering Comittee will review concept documents, development plans, and budgets to insure that they are consistent with the objectives of the participating organizations. If Steering Committee policies must be interpreted, and there is insufficient time to coordinate with the members, the Chairman is empowered to act for the Steering Committee. In such cases, the decisions will be documented and coordinated with other Steering Comittee members at the earliest possible date. 4a2a

2. PROJECT MANAGEMENT OFFICE - AFDSDC 4a3

The Project Management Office will coordinate the multiple activities required to make the NSW a viable system. These activities include system integration of extensive software/hardware deliverables from multiple contractors, acquisition and integration of tools for the expected software development centers, operation of the NSW, and the conduction of experiments with users to determine software productivity enhancements. The Project Management Office will manage the financial and budgetary aspects of the project, coordinate the efforts of the various Offices and Contractors, and provide a point of contact for questions from Management and external organizations. The Project Management Officer will be the individual tasked with these responsibilities. The PMO will be provided by AFDSDC. He will be a field-grade officer (or civilian equivalent) and will have a full-time staff of three to four people. His areas of responsibility include: 4a3a

Financial Management 4a3a1

The PMO will be responsible for the preparation and maintenance of the Project Budgets, for the tracking of expendatures, and for highlighting opportunities which require additional or diverted funding. The PMO will review budgets and plans of related projects for

opportunities for joint funding, and to insure neither "gaps" or "overlaps" exist. Such project financial information will be distributed by the PMO as directed by the Steering Committee. 4a3a1a

Technical management and review of system evolution 4a3a2

Maintain close contact with Technical Coordination contractor. Keep complete project milestone projections and history files. Monitor completion of milestones, both with respect to date completed and valid delivery of systems which meet product specifications. Coordinate/prepare proposals for PMO to minimize impact of any projected slippages. 4a3a2a

Prepare and/or compile concept documentation for PMO and Contractors. 4a3a2b

Review and evaluate contractor-prepared detailed product specifications to insure operational requirements will be met. Propose changes as appropriate. 4a3a2c

Coordinate contractor efforts; facilitate and monitor inter-contractor communication. 4a3a2d

Arrange and support Conceptual and Design reviews by Steering Committee, Advisory group, and/or selected external agencies at appropriate points in development of NSW total system, and of key component subsystems. 4a3a2e

Document and distribute results of all reviews, complaints/suggestions for technical improvements, and other inputs to conceptional/technical model of NSW. 4a3a2f

Explore and propose extensions to current design concepts to improve NSW's user interface, cost effectiveness, efficiency, flexibility, generality and/or reliability in the operational environments of current and projected users. 4a3a2g

Capture best versions of source and object code, and procedures for loading and executing same, prior to end of contractual periods (to insure continuity of project). 4a3a2h

Configuration Selection 4a3a3

Prepare (or supervise preparation of) technical and policy evaluations of distributed vs. regionalized vs. centralized location of Tool Bearing Hosts. 4a3a3a

Prepare (or supervise preparation of) RFP and/or sole-source hardware specifications for Framework computers, Frontend computers, standard terminal configurations, any other large scale hardware procurements required to implement or expand production system. 4a3a3b

Prepare, update, and distribute planning guidelines for configuration requirements to support different classes and levels of work within NSW. 4a3a3c

Prepare configuration requirements for AFDSDC projected expansion plans. 4a3a3c1

Provide assistance to AFDSC, RADC in preparing configuration plans to meet workload requirements. 4a3a3c2

Assist potential users in developing configuration requirements. 4a3a3c3

Release Control and Acceptance Testing. 4a3a4

Develop procedures for testing/verifying new versions of Framework and basic support tools prior to release for general usage. 4a3a4a

Conduct pre-release Environmental System Tests of new releases of Framework, basic support tools. Report results to Steering Committee. 4a3a4b

System Documentation. 4a3a5

Develop requirements and standards for system and user documentation, disseminate as appropriate. 4a3a5a

Maintain a complete and up-to-date library of system and user documentation. 4a3a5b

Monitor Contractors to insure documentation produced is consistent, complete, and in accordance with contract requirements and standards. 4a3a5c

Facilitate Planning and Policy Formulation 4a3a6

The PMO will be responsible for researching, documenting and preparing proposed positions for Steering Committee approval on planning and policy issues. In cases where differences of opinion exist among various project personnel/organizations, the PMO will be responsible for collecting and summarizing position papers representing all viewpoints. 4a3a6a

The PMO will select, with Steering Committee concurrence, an NSW Measurements and Evaluation Office and/or contractor(s), and direct such efforts as are necessary to evaluate and improve the cost effectiveness of the total system. 4a3a6b

Management and External Interface 4a3a7

The PMO will maintain overall project visibility and answer (or task the appropriate individual(s) to answer) questions pertaining to the project, both from the management of the participating Organizations, and from external agencies. The PMO will actively seek out other Air Force and DOD projects which could be useful to the NSW effort, or which should be aware of the the goals and progress of the NSW. 4a3a7a

2. TOOL SELECTION, ACQUISITION, AND INTEGRATION OFFICE - RADC 4a4

Introduction 4a4a

As in other industries, the productivity of the programmer in a software factory is related to the sophistication of his "tool kit". It is current practice for programing shops to build up a set of tools matched to the needs of their own applications and programmers. 4a4a1

In the National Software Works, the Works Manager and Front-end provide a consistent file system and user interface to a set of software programing tools which are distributed on a computer network. A network-based software production environment is an improvement over current methods in two ways: it will increase the sharing of tools among projects and organizations and it will remove some of the physical constraints which have prevented adequate tool kits from being available for many DoD computers. 4a4a2

The long-term goal is to have the NSW be a competitive marketplace for software development and maintenance

tools, with each installation independently selecting the tools which best meet its requirements. RADC is responsible for insuring that a critical mass of tools are available to meet the requirements of the initial users. RADC is also responsible for proposing standards and procedures to control the installation and maintenance of tools in the future. 4a4a3

Responsibilities 4a4b

Software Development Tool Index 4a4b1

An index of available software development tools will be compiled which will eventually represent the sum total of applicable tools owned by contractors and research centers. Characteristics of each tool such as its host, its programing language, its function, file input/output requirements, limitations, and cost of execution will be outlined. Requests from users concerning specific tool existence can be handled by reference to the tool index which will eventually be on-line to NSW users. Selected tools can then be procured, modified, and installed in an appropriate IBH for the NSW user. An index of NSW tools will also be maintained. Specific classes of tools include the following: 4a4b1a

Language processors- compilers, preprocessors 4a4b1a1

Debuggers 4a4b1a2

Management- programing support libraries 4a4b1a3

Measurement 4a4b1a4

Documentation 4a4b1a5

Design Aids- simulators 4a4b1a6

NSW Tool Installation 4a4b2

RADC will be responsible for publishing and maintaining tool installation guidelines which specify requirements for converting ordinary tools to NSW tools. Initially, these guidelines will originate from the major contractors of the NSW core software. Technical expertise on tool installation that is gained locally as tools for RADC IBH's are integrated

within the NSW will be offered to other tool installers. RADC will offer specific aid to tool installers through in-house efforts or by managing the procurement of services for tool integration. 4a4b2a

As each new tool is installed, it will be validated through this office to ensure conformance to an established set of standards which reflect proper interfaces to FE, WM, and the TBH. 4a4b2b

Tool Kit Selection 4a4b3

It is anticipated that NSW users will be grouped in areas of programing activity. Each group will require its own programing environment and tool kit which is related to the type of activity of the site. For example, programmers of communications software may have a different tool kit then ones writing base level COBOL software. There will also be overlapping tools such as editers used by both groups. 4a4b3a

The first cluster of programmers will emerge at the Data Systems Design Center where an estimated 700 programmers will be on-line to the NSW within six years. A second cluster will emerge at the Data Services Center. As the advantages of using the NSW become apparent to the programmer community, it is expected that many other groups will begin to use the NSW. 4a4b3b

It is the responsibility of RADC to interact with potential NSW user groups and ascertain their particular programing development tool requirements. Working closely with the user groups in an advisary role, RADC will suggest those tools which are most applicable and available. 4a4b3c

RADC TBH Installation 4a4b4

As a research center in the development of software tools, RADC has a vested interest in becoming an active member of the NSW community through the installation of our computers as tool bearing hosts. Currently, it is planned to augment our MULTICS and GCOS computer systems to become TBH's and install a subset of our existing tools in the NSW environment to conduct experiments. Technical expertise will be gained through such installation which will reflect in

a growing knowledge of general tool integration and will aid in closing the loop in our role as tool installation advisors. 4a4b4a

The tools which RADC will provide will be in support of a number of experiment which are currently being planned in the following areas which have been detailed earlier in the Plan. 4a4b4b

Programer Productivity Measurements 4a4b5

It has long been desired to determine the effect on programing efficiency of a proper and adequate set of tools and a discipline for programing. A number of experiments are being planned which attempt to measure this process and answer many of the questions that have been posed. The effect of structured programing techniques with programmer support libraries on production efficiency, readibility, and maintainability of produced software will be determined. The NSW and its tool environment can provide an ideal environment for conducting such tests. 4a4b5a

Language Control Facility 4a4b6

An attempt will be made to determine the effect of networking on the problem of maintaining computer languages. A centralized facility with tools for compiler specification, generation, and maintenance of compilers which are available to a wide range of users should tend to stabilize those languages. Changes due to language evolution will be made centrally resulting in fewer dialects. 4a4b6a

4. NSW OPERATIONS OFFICE - RADC 4a5

NSW Operations 4a5a

The goal is to develop a stable, reliable, responsive and cost-effective service to NSW users. Since the NSW is the first of a class of distributed systems, aimed specifically at augmenting the programing environment of AF organizations and their contractors, special consideration must be given to its operation while it is still in development. Development funds and effort must be applied to devising operational policies and

procedures if the NSW is to successfully make the transition from a R&D project to an operational system. 4a5a1

Constraints 4a5b

There will be parallel operation and development within the NSW world. Parts of the NSW and versions of the parts will be considered operational at any point in time. Criteria for deciding when a component or version moves from developmental to operational status must be clearly defined. 4a5b1

Although the underlying mechanisms of the NSW should be almost "invisible" to the user, they must be highly visible to the operators and maintainers. Several levels of documentation on the system, its components and its protocols must be developed and maintained to support system management, maintenance, tool installation, training and trouble shooting. 4a5b2

An environment must be created where tools can be readily added, deleted and modified to meet the needs of a growing NSW user community. 4a5b3

Guidelines, standards and ultimately certification procedures must be developed to allow the orderly modification of the core system and the addition of new tools and tool bearing hosts. 4a5b3a

The financial manipulations necessary to purchase the access to tools and computer resources should also be as painless as possible. Contractual mechanisims must be established to provide for acquisition, accounting and billing for computer resources. 4a5b4

Mechanisms have to be established to handle problems on a real-time basis and to provide feedback to system developers on efficiency and effectiveness. 4a5b5

Approach 4a5c

The approach in dealing with the above issues and bringing the NSW into the AF inventory will be to establish a NSW Operations Center (NOC) at RADC. RADC will be supported by ISI in this activity. After resolution of the key issues, establishment of operating policy and shakedown of procedures, the management of the

NOC will be turned over to an operational AF or DoD organization. 4a5c1

The NOC will use that portion of the NSW that is operational at any instant in time to perform its functions. The policy and procedures developed should therfore be based on practical experience rather than on intuitive speculation. 4a5c1a

RADC will act as a broker, in planning for and procuring computer resources, training and documentation. They will monitor resource useage via the WM and issue the necessary invoices, and perform the contractual paperwork necessary to meet invoices from TBH and tool suppliers. 4a5c1b

RADC has accepted this responsibility and is qualified to manage the NSW operations because of: 4a5c2

experience with the ARPANET and NLS 4a5c2a

experience gained in setting up the WUS 4a5c2b

contractual focal point for NSW development 4a5c2c

contact with other AFSC S/W development projects 4a5c2d

within the mission to do "advanced development...fine tuning, engineering, cost effectiveness" 4a5c2e

TASKS 4a5d

The following tasks need to be accomplished during FY76 to place the NSW in a position where it can begin operation. 4a5d1

Documentation 4a5d1a

establish a NSW documentation framework. 4a5d1a1

develop a descriptive NSW systems document. 4a5d1a2

develop a WM functional description document. 4a5d1a2a

develop a FE functional description document. 4a5d1a2b

develop a protocol functional description document. 4a5d1a2c

develop a TBH specification document. 4a5d1a3

develop a tool installation guide. 4a5d1a4

develop a PMT functional description document. 4a5d1a5

User guides 4a5d1a6

develop a NSW userguide. 4a5d1a6a

update the NLS userguide. 4a5d1a6b

Help data bases 4a5d1a7

update the NLS help data base. 4a5d1a7a

develop a WM help data base. 4a5d1a7b

Computer Resources--TENEX and MULTICS for the first year. 4a5d1b

develop an integrated plan for computer resources required to support DSDC, DSC and RADC. 4a5d1b1

establish contractual procedures for acquiring the necessary resources. 4a5d1b2

determine an equitable means of distributing NSW overhead. 4a5d1b2a

establish contractual procedures for "automatic" billing and payment. 4a5d1b3

establish WM resource useage reports (detail, format and frequency). 4a5d1b4

maintain historical resource useage record for future expenditure rate estimation. 4a5d1b5

Training 4a5d1c

establish course material and conduct training sessions in use of NSW and NLS. 4a5d1c1

establish syllabus for training trainers at DSDC, DSC and RADC. 4a5d1c2

define and implement basic lessons for NSW and NLS in SCHOLAR. 4a5d1c3

Trouble shooting 4a5d1d

establish NOC feeback capability for accumulating, classifying and analyzing problems and responses. 4a5d1d1

Install WATS lines between NOC, the users and the principle NSW developers. 4a5d1d2

establish fault isolation procedures. 4a5d1d3

Contractual 4a5d1e

interact with procurement to establish a smooth money transfer mechanism where there is: 4a5d1e1

an NSW overhead that must be equitably distributed across users, 4a5d1e1a

multiple suppliers of TBH's and tools, 4a5d1e1b

with a mix of GFE, not for profit and profit, 4a5d1e1b1

multiple users of subsets of these tools, 4a5d1e1c

where the exact user-supplier matrix cannot be completely determined ahead of time. 4a5d1e1d

5. NSW MEASUREMENT AND ANALYSIS OFFICE - Not Assigned 4a6

There is a continuing need for an office charged with the responsibility of providing techniques and manpower to measure the effect of using the tools and facilities provided by the NSW upon the individual users and the organizations of which they are a part. An Office of Measurements and Evaluation will be formed to fill this need. At this time, its functions are relatively clear, but an organization with the expertise and experience to fulfill this responsibility has yet to be selected. One of the first duties of the Program Management Officer will be to prepare a proposal for Steering Committee approval, designating the organization selected to be the Meaurement and Evaluation Office. 4a6a

Tasks: 4a6b

The Measurements and Evaluation office will be charged with the following responsibilities: 4a6b1

To develop measures and procedures for evaluating productivity of data automation personnel. This includes at least the following catagories : 4a6b2

System Analysts 4a6b2a

Development Programmers 4a6b2b

Maintenance Programmers 4a6b2c

Documentors 4a6b2d

Administrative Support Personnel 4a6b2e

(Project Managers?) 4a6b2f

To select, evaluate, and prepare data on productivity of representative groups of such personnel at AFDSDC, both with and without NSW. Care must be taken in the handling of these tests to insure no bias is allowed to creep in, as the results will bear heavily on both the future direction and speed at which the project moves. 4a6b3

To prepare an evaluation on the economic costs and benefits of NSW support on the test group, and to extrapolate as closely as possible the economic trade-offs as applied to the full Center. The evaluation should be briefed to both AFDSDC and NSW management. 4a6b4

To perform, at a somewhat later date, a similiar analysis of AFDSC use of NSW 4a6b5

To recommend, where appropriate, changes in the concepts of operation to improve the cost effectiveness of the NSW operation. 4a6b6

To document the methods and results in detail, so that similiar tests can be carried out by new members of the NSW community as they join. 4a6b7

To serve as a central clearinghouse of techniques and results of productivity measurements in the software development environment. 4a6b8

To maintain constant visibility over the success of NSW

efforts to increase productivity; to perform additional testing where appropriate as new features and facilities are added to the NSW; to propose and implement new procedures and measures as appropriate. 4a6b9

To gather and evaluate user and management reaction pertaining to non- quantative measures of project success, including user and management reaction to the system, changes in work patterns, job satisfaction, etc. 4a6b10

Interim Operation 4b

RADC, as one of the operational organizations participating in the development effort, has been tasked take responsiblity for the interim operation. The Works Manager, NLS, and several other tools will run on TENEX computers. Requirements have been identified for GCOS, MULTICS, and IBM 360/370 development machines. Additional B4700 capacity is also needed. The IBM 360/370 requirement and the B4700 requirement can both be satisfied by adding SADSC to the ARPANET. For the 360 requirement, there is also an option of connecting the machine at Eglin AFB. 4b1

The RADC MULTICS is the logical choice to provide an unclassified MULTICS development environment. MIT can be used for backup and access to special software. During FY76, TENEX computing will be supported at ISI. Office-1 will also be used for text processing, although that work will not be under the control of the NSW Works Manager. 4b2

NSW Plan

(J33112) 31-JUL-75 07:08;;;; Title: Author(s): Mike A. Wingfield/MAW; Distribution: /MAW([ACTION]) ; Sub-Collections: NIC; Clerk: MAW; Origin: < WINGFIELD, PLAN.NLS;4, >, 31-JUL-75 06:45 MAW ;;;;####;

1000-1hr arrive- Visit with Col Baggiano, meet maj hearn, mrs simmsBld 325 1

Maj Hearn will be the new Project Manager. Mrs Sims will be working with me, and probably take over a good deal of the 'user assistance' types of things I am doing now. This will be in Bld 325, which is where we have always been, the tour should start here. 1a

45min- Stalog- Maj Woody Gunter Industrial Park leased facility 2

The people in stalog are currently heavy users of NLS, and have been one of the prime pushers of things in NSW for quite a wile. They are currently using NLS to maintain and manipulate a project information file on the stalog project. They are also the ones who wanted VIMCCS, and DOSS. They are also considering the strong posibility of continuing to use NLS/NSW to maintain AFM66-1, if Air staff should move it down here (seems likely). They also have some interest in cross compilers, translators, graphics, and anything els whichseems 'Geewhiz'ish. Had some contact with RAND re Intelegent Terminal Project. Have a copu of a handheld terminal they are evaluation for use as an entry device for logistics info in wharehouses, etc, DO NOT MENTION PACKET RADIO TO THEM: THEY WILL WANT THREE COPIES NEXT WEEK! 2a

30 min- LGS(Mr Cooper)U1050 GIP 3

You need to touch base with Mr Cooper and if possible, maj Sawyer, his boss, to show the colors, and discuss just what you are trying to do ad what success you are having with the U1050 emulator. 3a

45 min- lunchC Club/wit Col B or Mr Mayhan/Mr Clogsdon 4

Man does not live by bread alone 4a

30 min- SCSA- Field assistance banch, Sge CrabtreeBlockhouse/Bld 857 first floor center 5

Sgt Crabtree is in the proces of starting to use NlS for management of the Dificulty Reports and System Advisory Notices concerning the base level Standard systems. This Needs to be an early part of NSW, as these are what generate the major portion of our Maintenance workload. 5a

30 min- SCCR- Release control Branch, Mr FisherBlockhouse first floorright rear 6

Mr Fisher's branch is in charge of editing, printing, and

distribution of the manuals which come out of afdsdc. Is cautiously enthusiastic about NSW, but keeps us honest. 6a

30 min- SGX- Maj Green/Amn Loweblockhouse fourth floor 7

Maj Green and his people are interested in programing aids for the 360 (it appears the TRIMIS hospital will end up with one), graphics and medical systems available for evaluation/training/demo through arpanet, and the possibility of using our elf as a port to connect a terminal controler system to a major computer in a manner very reminicent of a combo of the Advance Video terminal system we once talked about mixec with a 'frame/form' type NSW PCP Front End. 7a

30 min- PRE -mrs wagnerBld 205, room 14 8

PRE will be the first users of NLS for production documentation, starting in the mid august timeframe. These people are typical of the people who will be using NSW for documentation. 8a

1515-leave for airport 9

If some of the above meetings cant get set up, the followingare some alternates which would be usefull: 10

SYOS/Lt Price- These will be the first users of NSW for B3500 programming, also are interested in using net resources for simulation, but UCLA's time is too expensive. If you get a chance, talk about the New SCERT/76 system, instead of having to all run on a 360 as before, the new system is modular, and all but one module can run n any machine with ansi cobol. this sounds like a good possible prototype for a distributed tool consisting of a canned set of systems running serially on different machines. 10a

XF/Capt Wolfe- has the thankless task of publisning the Basetop DPP, will probably use NLS, allowing him to turn what would have been a major disaster (in recards to when it is due out) into a minor disaster. This application really scares me cause nobody but him and mee seem to comprehend the scale of disaster they will have if they tru to publish with typewriters. It is reasonabbly certain that even if they use NLSand we give them al the resources they need, they cant accomplish the job on schedule. 10b

Please excuse all the typos. I am on my way out the door on leave, talk at you next trip perhaps 11

%Larry 12

(J33113) 31-JUL-75 07:27;;;; Title: Author(s): Lawrence A. Crain/LAC; Distribution: /GUNTER([ACTION]) ; Sub-Collections: NIC GUNTER; Clerk: LAC; Origin: < CRAIN, AGENDA.NLS;5, >, 19-JUL-75 12:14 LAC ;;;;####;

33113 Distribution
Air Force Data Systems Design Center ,

USER SERVICES WEEKLY REPORT for Week of July 21 - 25

USER SERVICES WEEKLY REPORT for Week of July 21 - 25: 1

from JMB 1a

Wrote a commands branch with Dee to handle Dick Watson's journal mail daily. This took quite a while because I ran the whole thing before checking any of its parts: it worked the first time, but the second time, when he didn't have any new mail, it blew up (of course) and it took some time to put all three files back together again. More time was taken getting the thing to work when there's no new mail (have to have it insert dummy mail). 1a1

I go into all this detail because it shows two general issues about writing commands branches for users: (1) debug each of the steps before you run the whole branch, because it makes one very nervous to mislay a guy's whole author branch. (2) you can either make the branch do the most direct thing and specify to the user that she only run it under such-and-such conditions, or you can expand it to run whenever she has that particular need with very few restrictions. I have found that nearly all commands branches I've written have to have preconditions. 1a1a

I may be expanding Watson's branch later, if/when he likes the current procedure and is ready for more, to combine the journal mail with his messages, which he has another commands branch put into NLS files. 1a1b

Learned more about commands branches: how to make them accept input from the user if you know how may characters she will type. You have to have the branch load a program that is a procedure-replace, and then you put a percent sign in the command for each character the user will supply. 1a2

Worked with SGR & PKA & PAW2 on the Edit Statement command, trying to see how it works (and doesn't work). 1a3

Helped Priscilla work on the Viewgraphs and attended meeting to approve them. 1a4

Worked on Content analyzers (susan's session) and user-programming a little. 1a5

Proofread JHB's response to Betty Finney's paper on text editors. 1a6

Fixed up help and the Brief Guide to User programs to document the new command Modify subsystem will have--Substitute spaces. 1a7

USER SERVICES WEEKLY REPORT for Week of July 21 - 25

from SGR 1b

Did a lot of talking to people my first day back from Alaska to catch up on things. Talked to Betty Finney, Rita, Jim N, Elizabeth, etc, to try to figure out what to do about the "mess" at the Pentagon. Rita's trip here was cancelled - she will probably come here the second or third week in August instead. 1b1

Actually this week involved a lot of talking... Talked with JHB about general training issues and also on the seminar planned for the last week of August. Talked with NDM on how to teach the Content Analyzer and then taught it to Pam, Priscilla, Jeanne, Dee, and Adrian with Dean as consultant. 1b2

Reviewed the viewgraph drafts prepared by Priscilla and after she made some minor revisions sat in the meeting for the approval of the drafts with US people, JHB and BEV. Priscilla and I then talked with our Report Services coordinator about actually getting the work done. 1b3

Worked on edit statement, and found a bug in the message program which has now been fixed. 1b4

Went through the data base used to produce courses and did a preliminary chop job for the seminar in August. Went to a meeting of selected Applications people where the course and other things were discussed. 1b5

Proofed an NLS User's Guide produced by Mil Jernigan for content and will pass it on to JHB. 1b6

from RH 1c

I started out the week heading for California but got sick in the interim and spent 2 1/2 days out sick. The last part of the week I spent trying to get ahold of people like Betty Finney, Liz Riddle, etc. which was a problem, and I spent most of the time at ARPA. On Thrusday there was a meeting on the XGP with Steve Crocker from ISI which was interesting since it dealt with the new changes they hope to bring off in the next few months. I also, (like everyone else it seems), worked on the Edit Statement command. On Friday I attended another meeting of the Steering Committee at SRI. 1c1

from PKA 1d

Monday- Worked on the June feedback statistics. Have found from the past three months that this will take several hours.

USER SERVICES WEEKLY REPORT for Week of July 21 - 25

Had a meeting with Susan to inform her of what we did while she was away for the past 2 weeks. Spent an hour doing the us report for the last 2 weeks. Somehow they just don't seem to get done on time... 1d1

Tuesday-Dee came in this morning about something or other, and ultimately went away with a commands branch for rww's mail and messages. Jeanne Beck wrote this and I followed the entire process carefully, because I don't have any experience with this. It really took all day to get the thing going. 1d2

Wednesday-Had a class on using content analyzers that was taught by Susan. This expanded my knowledge immensely. 1d3

Thursday-worked with Sandy on Feedback. We started from the very beginning. First running the commands branches that put the new messages into a form that can be worked with easily. Sandy answered one or two while I took note of the steps involved. Also got some info about who to ask about what when that is necessary. 1d4

Viewgraph meeting today with US and jhb and bev. 1d5

Misc: Learned edit statement command by talking and practice. I think it's kind of fun. Spent some time every day reading mail and organizing it. 1d6

from PAW2 1e

All day Monday and Tuesday finished the cutting and pasting of the viewgraphs. Tuesday afternoon went over with Susan her suggestions for revisions on the graphs. Wednesday morning attempted to clear up my initial file and read some of my back mail. That afternoon met with Susan to go over content analyzer patterns. They seem somewhat clearer at this point but I need to work with some to get a better idea of how they work. Thursday played around some with the Edit Statement Command and prepared for the viewgraph meeting scheduled for Friday. Friday morning met with all trainers, Jim Bair, and Bev Boli to discuss the viewgraphs. Spent Friday afternoon making the necessary changes and then Susan and I took them down to Toni Luft to discuss layout and production. They should be completed in time for us to take them to Gunter the first week in August. 1e1

USER SERVICES WEEKLY REPORT for Week of July 21 - 25

(J33114) 31-JUL-75 09:11;;;; Title: Author(s): Jeanne M. Beck, Susan Gail Roetter, Rita Hysmith, Pamela K. Allen, Priscilla A. Wold/JMB SGR RH PKA PAW2; Distribution: /JCN([INFO-ONLY]) ; Sub-Collections: SRI-ARC; Clerk: RH;

Your use of the term teleconferencing

Although I am in general agreement ith your paper on teleconferencing in NLS, I suspect that there may be some confusion at the conference on your use of the word "teleconferencing." Unless I am mistaken, you are using the word in a different way than Turoff and other developers of tleconferencing systems. Here is why I think confusion may arise, and some points you may wish to make in case it does arise. 1

Basically, the term teleconferencing (or rather computer teleconferencing) has come to mean a discussion: 2

1) among a limited group of people, 2a

2) for a limited period of time, and 2b

3) on a limited topic. 2c

A teleconferencing system, then, is a set of tools designed to simulate a group of people in a conference sitting around a table. 3

Under this use of "teleconferencing," neither the Journal system nor shared screen telephoning would normally be considered teleconferencing. Shared screen telephoning, for example, is one-to-one, not a group conference, and the Journal system does not provide a coherent set of tools for focusing a discussion. Both of these comments, however, need to be addressed more closely. 4

If shared screen telephoning is done on a room-to-room basis, i.e., if wall projection in distant conference rooms is used to tie together a group of people, this would be a teleconferencing application in the normal sense of the word. 5

Similarly, if a special directory is set up for a conference, then sendmail messages can be sent to the conference ident. The resultant journal branch substructure would then be a conference record in the normal sense of the word. However even this would be a marginal teleconferencing system; most computer cnferencing systems also provide pointers for each conferee, to allow them to access only "new" (for them) entires. Perhaps we should experiment with this form of teleconferencing ourselves, to see if features present in other computer conferencing systems should be built into a special subsystem. 6

Although NLS does not offer what might be called a "classical conferencing system, we do offer a sophisticated communication environmet that is adequate for many organizational communication uses. It seems to me that your paper is trying to broaden the meaning of teleconferencing, to include what NLS does. I see nothing

Your use of the term teleconferencing

wrong with that, but I think it would help if you noted your intention to broaden the term. 7

Your use of the term teleconferencing

(J33115) 31-JUL-75 11:16;;;; Title: Author(s): Raymond R. Panko/RA3Y; Distribution: /DCE([ACTION]) ; Sub-Collections: SRI-ARC; Clerk: RA3Y; Origin: < PANKO, AROSEISAROSE.NLS;1, >, 31-JUL-75 10:58 RA3Y ;;;;####;

JCP 31-JUL-75 14:03 33117

IN RESPONSE TO HGL'S 26210

NO, NO, A THOUSAND TIMES NO. TERMINAL TYPE TI733 IS NOT WHAT YOU SHOULD TYPE BESIDES "NO RAISE". IF YOU DO THAT, YOU'LL GET EXTRA UNNEEDED PADDING OVER THE NET WHICH MAKES EVERYTHING EVEN SLOWER. TERMINAL TYPE NVT IS ALMOST CORRECT, BUT ALSO HAS CERTAIN PROBLEMS. IN ORDER TO SET YOURSELF UP RIGHT FOR TNLS, YOU NEED TO TYPE "TSET<CR>" AT THE EXEC BEFORE YOU GO INTO NLS. TSET IS A LITTLE HACK I INVENTED BECAUSE I WAS TIRED OF TYPING IN HALF A DOZENDUMB THINGS TO GET MY TERMINAL RIGHT FOR TNLS. "TSET" IS AVAILABLE ON ALL THE SYSTEMS WE USE AND CAN BECALLED DIRECTLY FROM THE EXEC. THIS TAKES ONLY A SECOND, AND DOES LOTS OF GOOD THINGS TO YOUR TERMINAL SPECS TO MAKE EVERYTHING OPTIMAL FOR TNLS USAGE. SO NEXT TIME, JUST TYPE TSET AT THE EXEC BEFORE YOU GOINTO TNLS, AND DON'T BOTHER WITH "NO RAISE" OR ANY OF THAT OTHER GARBAGE LIKE "TERMINAL TYPE IS TI733. ENJOY - JEFF

1

JCP 31-JUL-75 14:03 33117

IN RESPONSE TO HGL'S 26210

(J33117) 31-JUL-75 14:03;;;; Title: Author(s): Jeffrey C. Peters/JCP; Distribution: /SRI-ARC([INFO-ONLY]) ; Sub-Collections: SRI-ARC; Clerk: JCP;

33117 Distribution

Douglas C. Engelbart, Martin E. Hardy, J. D. Hopper, Charles H. Irby, Harvey G. Lehtman, James C. Norton, Jeffrey C. Peters, Dirk H. Van Nouhuys, Kenneth E. (Ken) Victor, Richard W. Watson, Don I. Andrews, Mary Ann Kellan, Buddie J. Pine, Andy Poggio, David L. Retz, Laura J. Metzger, Karolyn J. Martin, Jan A. Cornish, Larry L. Garlick, Priscilla A. Wold, Pamela K. Allen, Delorse M. Brooks, Beverly Boli, Rita Hysmith, Log Augmentation, Joseph L. Ehardt, Raymond R. Panko, Susan Gail Roetter, Robert Louis Belleville, Rene C. Ochoa, Ann Weinberg, Joan Hamilton, Adrian C. McGinnis, Robert S. Ratner, David S. Maynard, Robert N. Lieberman, Sandy L. Johnson, James H. Bair, Jeanne M. Leavitt, Rodney A. Bondurant, Jeanne M. Beck, Marcia L. Keeney, Elizabeth K. Michael, Jonathan B. Postel, Elizabeth J. Feinler, Kirk E. Kelley, N. Dean Meyer, James E. (Jim) White

MIKE 1-AUG-75 07:14 33118

New Orleans Paper : Trends in Teleconferencing and Computer-Augmented Management Systems

For you review....the manuscript is on the way. The actual talk will stress the results of the evaluation to date (December 2, 1975).

Trends in Teleconferencing and Computer-Augmented Management Systems 1

Michael T. Bedford
Supervisor - Business Planning Group
Bell Canada
Montreal, Quebec, Canada 1a

Introduction 2

This paper presents a general overview of the different aspects of teleconferencing (audio, audio-visual, graphic, and data) and then shows how the last last of these - data, or computer conferencing - can be used to facilitate a number of group communication functions which can not be achieved through the other media. 2a

Following a discussion of the group communications or interpersonal aspects of computer conferencing, a bridge is made to the concept of an intra-personal computer conference - a conference with only one participant. Such "conferences" constitute a fundamental building block of the office automation systems of the future. 2b

The contents of this paper are based on the research and on-hands experience of Bell Canada's Business Planning Group. This group is Bell Canada's long-range market planning entity, and is responsible for identifying new business opportunities and threats to existing markets. Technology forecasting, technology assessment, and policy analysis constitute major group roles. The group also provides consulting services to outside clients in areas of methodology development and applications, technology assessment, and policy formulation. 2c

Teleconferencing - Meetings Over Distance 3

Teleconferencing Overview 3a

The Business Planning Group has conducted a number of studies investigating the interrelationships between travel and communications on both a macro- and micro- basis during the past three years. The investigation has considered both the substitution and the supplementation elasticities of travel and communications (including the reasons for preferring one over the other), as well as the energy trade-offs involved. Both inter-city and intra-city transportation were included. 3a1

The substitution of communications for travel introduces the notion of the teleconference - a conference held at a distance. With the aid of teleconferencing media, a group of

geographically dispersed individuals can occupy a single information space. The "size" of that information space (to borrow one of Gordon Thompson's three dimensions of communications utility) may be quite large, thus providing the participants with a feeling of "sharing" that space with the other participants, or it may be quite small, giving the participant the feeling that he is cut off from the other particpants. Our research indicates that the "size" of the information space provided by a teleconferencing medium (or any communications medium, for that matter) is critical to the amount of information (useful data) exchanged, but more importantly (and this is surprising to some), the size of the information space is not a simple function of the bandwidth of the medium, nor the speed of the transmission. 3a2

Our studies indicate (and common-sense would seem to confirm) that the size of the shared information space (and thus the participants' feelings of participating and belonging) is not so much a function of the total volume of data transmitted in a given time (ie. bandwidth), but rather the volume of useful information transmitted. A common assumption that trips many venturers is this field is that the way to provide the greatest sense of "belonging" to each member of a dispersed group is to simulate as closely as possible the face-to-face atmosphere they would "enjoy" had they chosen to travel rather than telecommunicate. While this assumption may be true in the extreme case (teleportation or perhaps tele-holography), it is definitely not true in the case limited by our current technical capabilities. 3a3

In trying to duplicate the face-to-face atmosphere in a technology- and bandwidth-constrained environment, many reseachers have been absorbed in a linear thought process, failing to make the lateral connection to a whole new approach to the problem of providing a feeling of particiption to the scattered individuals. This alternative approach does not try to duplicate the face-to-face atmosphere, but rather asks, how can the limitations of a face-to-face environment be reduced through the use of telecommunications media ? It will become clear in the following discussion that there are a number of alternatives to trying to duplicate a face-to-face atmosphere as a means of increasing the utility of the teleconference media, and that in addition to relying on much lower bandwidths than the face-to-face replicators, these alternatives provide significantly more powerful communications capabilities to participants. (In this latter area of measuring communications power, effectiveness and utilty, the Systems Engineering Group

of our research subsidiary, Bell-Northern Research Laboratory, has been invaluable.) 3a4

Teleconferencing Media 3b

Audio 3b1

To date audio conferencing systems have been of two types: group-to-group conferences, utilizing variations on the Bell System Speakerphone microphone-loudspeaker combination, or multi-point, operator-assisted calls, in which the phone lines of three or more calling parties can be bridged together. 3b1a

Audio conferencing is an attractive concept for telephone company marketing representatives, because it offers the prospect of additional revenues without substantial increases in operating plant. The main reasons for the failure of this type of service to have developed more completely have been attributed to a lack of promotion on the part of the phone companies, and a rate structure that appears confusing. 3b1b

Most audio conferencing facilities appear to offer customers good service for the money spent, but there are particular functions for which they are not feasibile. Obvious examples include the inability to cater to those "meetings" with a high graphic, pictorial, or numerical subject matter, difficulty in identifying the participants by voice, the impossiblity of providing for side discussions, whispered or otherwise, and difficulty in structuring the meeting throughout its duration. To a certain extent, these limitations can be overcome through the use of facsimile machines in each of the conference locations (or pre-mailed graphics packages), speaker identification systems, and built-in manual interupt devices, but these are neither comfortable or natural for the user to learn, nor co-oincident with the narrowly-defined concept of audio teleconferencing. 3b1c

Audio-Visual 3b2

The addition of a video component would appear to many to solve a number of the limitations found in audio-only systems. These video advocates point to the ability to share documents, to identify not only the individual speakers but relate to their non-verbal communications cues,

and the ability to communicate out of the mainstream of the formal meeting process. 3b2a

In fact, our research into video conferencing indicates that many of these claims are unfounded, and that video conferencing does not contribute in any significant way to the size of the information space shared by the conferees. 3b2b

Our research is based on the four-city, studio based conference television network that Bell Canada has made available free of charge to any business group wth an interest in trying out this new medium. There are studios in Bell premisses in Toronto, Ottawa, Montreal, and Quebec City. Each is equipped with a seating arrangement that puts up to eight members in an arc in front of three voice-switched cameras, and two studio monitors (or large rear-projection wall screens), carrying the incoming and outgoing signals. Each studio is also equipped with a document camera. The cameras in each studio are controlled from one of the seating positions in that studio. The 6 MHz signals are carried over the stand-by or protection channel that Bell Canada supplies to the Canadian Broadcasting Corporation. 3b2c

(We have also done some research into the feasibility of using Picturephone apparatus in a conferencing environment, but discontinued it when we found that the screen size was too small for group use, and the resolution was too low for meaningful document transmission.) 3b2d

Problems we have identified in relying on video conferencing for group meetings include the following: 3b2e

Difficulty in transmitting meaningful pictorial or graphic information; the resolution is quite low, the graphic camera is not interactive, and it is impossible to call up stored records (data or graphics). (The inefficiencies inherent in relying on a full video channel for transmission are a subset of the general high perceived cost of dedicating these wideband facilities to a specific service which is used only periodically.) 3b2e1

There is no record of the meeting produced or stored by the medium; videotapes have been used in the past, but these have been cumbersome and difficult to access later; transcriptions of what has taken place often fail to reflect the nature of the interaction, and in any case

are of use only to the extent that they are distributed and catalogued by the conferees. 3b2e2

There is no record of the meeting produced or stored by the medium; videotapes have been used in the past, but these have been cumbersome and difficult to access later; transcriptions of what has taken place often fail to reflect the nature of the interaction, and in any case are of use only to the extent that they are distributed and catalogued by the conferees. 3b2e3

The medium does not appear to produce more meaningful dialogue or relationships than do audio conferencing media; that is, the addition of the video component does not create a warmer or more friendly environment to the extent that more intense types of communication (bargaining, negotiating) could take place effectively. 3b2e4

A partial solution to some of these problems could be found by relying on slow-scan video for transmission of graphic information, and video compression techniques for reducing the system's consumption of wideband facilities. 3b2f

Graphic 3b3

In speaking of graphic conferencing, I am referring to the work that Bell Canada and Bell-Northern Research have done on Scribblephone, an interactive graphics terminal designed to facilitate the communications between two groups of people in remote locations. The first prototype Scribblephone was designed with one-to-one conferencing in mind, although the principle is extendable to group communications. 3b3a

The Scribblephone concept provides each party with an alpha-numeric keyboard, graphic CRT display, and a lightpen or similar device. It differs from the tele-blackboard concept in that it is interactive; what Party A writes on his screen appears on both screens; what he erases is deleted from both screens. This interactive feature greatly enhances the communications capabilities of individuals or groups with a need to communicate with other individuals. 3b3b

In the set of conferencing media discussed up to this point, Scribblephone is unique in so far as it is the first medium that can actually transcend the face-to-face alternative in terms of the size of the shared communications space, and the degree to which that space is shared. It is not

difficult to imagine two engineers, architects, or technicians using two Scribblephones side-by-side in the same room, in order to "get into" the same informaton space, which they could not do in a pencil and paper mode. The increase in distance from a few feet to a few miles to several hundred miles would not appreciably detract from their abilty to share the same information space. 3b3c

Providing the Scribblephones with local storage capability would permit them to quickly access documents, to record a transcript of a session, and possibly replay the session at a later date. 3b3d

Computer Conferencing 3b4

Computer conferencing is a facility which permits a number of dispersed (through either time or distance) individuals to exchange information, have the information automatically catalogued, and access the information at any time in the future via a variety of retrieval keys. Participants in the conferences use teletype-compatible terminals to enter messages for other members of the conference, receive messages put into the conference since their last sign-on, or take advantage of the computer's automatic cataloguing and storing of all previous messages by retrieving some or all of the previous entries in the conference). Computer conferences can last anywhere from a couple of days to many months. 3b4a

Such conferences are different from the mailbox systems associated with most timesharing machines. Normally, a participant in one conference will belong to many other conferences as well. When he signs on to the computer, he is given a message indicating which conferences have had activities since he last joined them. He can then join each of the conferences in turn, as he sees fit. Thus each conference is dedicated to a particular subject, and a particular membership; new members can join conferences in progress and be brought up to date rapidly on the communications to date, since each conference is actually a continuously updated log of all the entries to-date, wth the exception of private messages, which are eliminated from the record after the hve been received by the recipient(s). 3b4b

Computer conferencing is the only one of the conferencing mediums that permits non-real-time conferencing. In fact, computer conferencing excels in this situation, and has

limited appeal in situations where both parties wishing to communicate are free at the same time. 3b4c

Applications of computer conferencing that take best advantage of its capabilities are numerous: 3b4d

task forces composed of members of different plants or departments who cannot get together for face-to-face meetings without considerable advance notice, and yet who must communicate developing situations and reactions quickly. 3b4d1

joint authorship of papers; the authors can review each other's work-to-date, comment on it, and take it into consideration when writing their own material; 3b4d2

policy implementation and coordination in a decentralized environment; one of the first significant applications of computer conferencing was in the Office of Emergency Preparedness when it was responsible for coordinating the Nixon wage-price freeze program in 1972; 3b4d3

policy development when the unbiased opinions of a group of individuals is desired; computer conferences permit participants to add anonymous messages to the conference, thereby permitting conferees to voice opinions that may not be popular with others, or which might involve some form of embarassment if made in a face-to-face environment; 3b4d4

as aids in decision making, computer conferences can be designed to administer questionnaires to knowledgable people in a number of areas, tabulate there answers automatically, feedback the responses and collect opinions from the respondents on the pooled answers to the questionnaire; such on-line Delphi research was one of the instrumental forces in spurring the development of computer conferencing systems. This capability has been expanded to the point where computer conferences are adminsitering cross-impact analysis matrices and linear programming algorithms to groups of anonymous, geographically dispersed "experts" in different fields for a variety of applications. 3b4d5

Office Automation Systems - On-line Work Space for the Manager of the Future 4

The previous section on computer conferencing described how

researchers and administrators today are using on-line information storage and retrieval systems to build on-line computer conferences which they can use in communicating with other members in their field, and to maintain records of those communications in an orderly fashion for use by themselves or others at a later date. The office automation concepts referred to in this section refer to the use of a similar on-line storage capability to aid the manager or decision maker of the future (knowledge worker, in Drucker's words) in organizing his use of information and communications, and to facilitate the incorporation of that assimilated information into a format useful to others in his immediate corporate community or in other communities with similar managerial or operational interests. 4a

Future managers and knowledge workers will have the opportunity to compose all their documents at a computer terminal, with the developing document being stored in a time-sharing computer. Electronic cutting and pasting and appending sections of one file or work piece to another will permit the knowledge worker to get his thoughts documented much faster than currently possible. The computer can handle the production of hard-copy for the worker, producing either machine readable output (for storage on cassette tapes), local printing on conventional printers, or computer-output-microfilm for producing camera-ready masters with various type fonts, page widths, and graphics capability if needed. Work-directed communications between knowledge workers can take place through the computer, using its capabilities to automatically catalogue the correspondence and make it readily available if required in the future. Knowledge workers will no longer be tied to a particular physical location to do the knowledge work aspects of their job. Work spaces will be distributed in workers home, in community work centers, airports and schools. Portable terminals will permit knowledge workers to communicate with their colleagues, their personal information spaces, and their colleagues' information systems from anywhere they can access the telephone network. 4b

These office automation systems differ from the management information systems that have played a major role in the applications side of office computer literature for a number of years. The most significant differences include the following: 4c

They are designed to be used by managers, not by trained technicians or clerks for the use of managers. The background or training required to manipulate the information in the system reflects the fact that the manager will not be willing to invest a great deal of time in learning these routines. 4c1

Managers are provided with the ability to organize their information space in a manner that seems natural to them, rather than a predetermined manner that seemed natural to the system designer or a human factors analyst. Information can be organized in an unlimited number of formats, and no two people would be likely to organize their work spaces identically, since no two would have identical perceptions of the interrelationships between the different elements. 4c2

Despite the flexibility afforded in organizing the information, the structural guides available in such systems permit users to elaborate and develop ideas through presenting the information in different formats and comparing the differences implicit in the different formats, much as one would do with a list of ideas written on paper file cards which could be sorted in different order to convey different impressions. 4c3

The ability to structure the information space in a hierarchical fashion permits individuals who are not familiar with an author's method of file organization to enter that work space and find the information he seeks, assuming the author has organized the space with sub-elements of major thoughts nested "beneath" the upper levels of the hierarchy. (Failing this, the ability to scan a work space and see only the first part of each sentence or paragrap or chapter would still save he information seeker a great deal of time in his search.) 4c4

There are a great many assumptions built into the above descriptions, and Bell Canada's Business Planning Group is mking a concerted effort to validate them. We are currently involved in an analysis of our group's use of OFFICE-1, an office automation facility developed by Stanford Research Institute's Augmentation Research Center (Douglas C. Engelbart, Director) and running at the Tymeshare site in Cupertino, California. (OFFICE-1 is connected to the ARPAnet, but Bell Canada access is through a multiplexed private line between Montreal and Cupertino.) 4d

The Bell Canada evaluation consists of several modules, each generating a specitic report. The representative modules include evaluations of the following: 4e

user attitudes to the computer system and the new work environment; 4e1

changes in users' work styles; 4e2

changes in users' managerial styles; 4e3

relative efficiency of our use of OFFICE-1 resources, compared to other users; 4e4

the future of OFFICE-1-type systems within institutions like Bell Canada; 4e5

the potential of such systems to generate revenue for Bell Canada, should they develop into widespread usage. 4e6

MIKE 1-AUG-75 07:14 33118

New Orleans Paper : Trends in Teleconferencing and Computer-Augmented Management Systems

(J33118) 1-AUG-75 07:14;;;; Title: Author(s): Michael T. Bedford/MIKE; Distribution: /LHD([INFO-ONLY]) ; Sub-Collections: NIC; Clerk: MIKE;

com 8/1/75

tape114 mta1, glossary--kp5s from this and prev. tape 112 thru p. 116. 1

FEED 1-AUG-75 15:09 33120

com 8/1/75

(J33120) 1-AUG-75 15:09;;;; Title: Author(s): Special Jhb Feedback/FEED; Distribution: /DMB([ACTION] dpcs notebook please, dee, thanks, sandy) &DPCS([INFO-ONLY]) ; Sub-Collections: SRI-ARC DPCS; Clerk: FEED;

33120 Distribution
Delorse M. Brooks, Documentation Production and Control System Interest Group ,

DVN 1-AUG-75 16:50 33121

Fourth DRAFT DDPCS Promotional Document

Following (33059,) Dave, Norm, Pat, Doug, and I met. The sense of the meeting was to follow essentially the plan described in the introdcution to the DDPCS Planning Documnet (33059,1a). Doug and Dave were to open contact with selected possible clients and I was to continue revising the promotion document which last appeared as (33006,). Here are my revisions. Note that some issues raised in square brackets below were not discussed and remain unresolved.

THE PROBLEM

1

The field of machine-aided document production is experiencing a period of chaotic growth. New hardware and systems raging from type writers with limited magenetic card memory through highly sophisticated systems such as SRI's Augmented Knowledge Workshop are appearing and disappearing from the marktplace. User's report startling successes and faileurs, but more frequently report uncertain outcomes in a field where the real costs of the old procedures are normally unknown, where organizational lines frequently inhibit change, and where the benifits as well as the problems of a new medium are frequently unforseen.* 1a

*[We need to expand this paragraph. I don't think I am the one to do it. Maybe Tom Humphry or Jack Bialik.] 1a1

At the same time the cost of coventional methods of publication is rising sharply. 1b

SRI'S CAPABILITY

2

SRI has been active in consultation and design of computer aided system for word processing and publication since 1962. The cumulative contract value of our experience is over N million dollars and work has been preformed for federal goverment, local goverment, the military and commercial clients. 2a

[] We have perfomred system analysis of machine-aided publication, considering in detail and choosing the most economical or efficient combination of procedures, hardware, and software, for a number of customers substantilly committed to computer-based document production . 2a1

[] In the aera of machine-aided publication and related computer-based information handling systems, SRI has proved out a design procedure in a large number of projects. 2a2

[] We have separately brought to prototype operation the Machine Aided Editing system, a mini-computer-based interactive, documentation production system. 2a3

[] We have developed the Augmented Knowledge Workshop, a related system based on time sharing and frequently used through computer networks. It is a highly interactive system which aides a wide variety of knowledge tasks, such as managment informaton flow, software system development. A

community of users exists where document production in is now a principal activity. 2a4

we believe that the present growing role of computers in documentation and increasing cost of convetional processing has brought to many organizations the problems we have been dealing with for some time. We are therefore seeking to expand our services to a wider circle of clients. 2b

METHOD OF OPERATION 3

First it is necesary to for SRI to understand the client's needs and present system sufficiently to propose the scope and character of the analysis and design work. In some cases the client has such information availalbe, in other cases SRI contracts for a preliminary investigation which takes about two man weeks. 3a

Following the primillinary investiagtion, SRI will propose anayslis, desing, and implementiation planning appropriate to the customer. The goal of the proposal will be a more effcient, economic, and rapid publication operation better integrated into the client's organization. The design phase is described further below. 3b

SRI's proposed work will lead to a report, face to face consutation, and suporting docuementation as necessary. As appropriate it may include anaylysis of the present sysetem, specification of hardware, a systems design, inplemetation planning, and turnkey operation. 3c

Some clients interested in maintinag touch with state-of-the-art development in this area my then choose to join the Documentation Development Community. 3d

FURTHER CAPABILITIES AND EXPERIENCE 4

Interdiciplinary Environment 4a

SRI's analysis of application of computers to publications has the advanatge that it takes place among the Information Sciences Group which is directly concerned with the development of improved analysis, design, and evaluation tecnniques. The group's work has a strong multidisciplinary character. It has pioneered the design of a range of large-scale data processing systems. Detailed knowledge is availalbe within the group of the current and projected state of the art for computer and communications hardware and software. It has worked in such

diverse areas as banking, transportation, services, education, process control, computer-aided design, and military operations. 4a1

Desing Procedure 4b

The Information Science Group has estabisihed and frequently applied a design process composed of three distinct phases: 4b1

In the first phase, the system goals of the user are translated into a set of realistic economic, technical, and procedural requirements. 4b1a

Such requirements become the basis for the second phase, overall system design. During the second phase the properties and interconnections of major system components are determined for items such as computer programs, input and output devices, data converters, memory devices, communication lines, and display devices. 4b1b

In the third and final phase, the performance of the proposed system is evaluated with respect to such factors as response time, accuracy, reliability, security, cost, and other factors. 4b1c

These three phases-requirements, design, and evaluation--compose a loop that may be traversed again, each iteration providing an increasingly refined system design. 4b1d

Project Experience 4c

Work on text processing and documentation systems has ranged from simple single-terminal systems to complex multiterminal editorial production systems operating within a distributed computer network. These efforts have encompassed a variety of facets of the process, including data capture; processing, editing, and formatting; and document output. 4c1

One notable project that has particular relevance here is the design of a complete editorial production system for Encyclopedia Britannica. SRI documented the entire editorial process for the Encylopedia Brittannica, prepared a preliminary system design for an machine-aided editorial system and performed an economic comparison of the proposed system versus the manual system. SRI delivered a final system design that included software, detailed hardware and software specifications, personnel requirements, schedules, and milestones. SRI also assisted with the initial system implementation phase. 4c2

[We need writeups on a few more projects. Maybe Tom Humphrey can provide them. 4c3

Machine Aided Editing System 4d

MAE is a minicomputer-based (PDP 11/20) text editing system developed in the Information Science Group. Its primary function is to provide an environment for the development of production-oriented text handling techniques and to demonstrate the application of these techniques to potential clients. A growing portion of MAE activity has been devoted to report preparation by SRI staff. For example, most reports generated by the Information Systems Group are processed through MAE, and other groups such as Chemical Information Services are working with ISG personnel to use MAE for production text handling. 4d1

MAE is a fairly powerful page-oriented text editing system. The approach has been to create an easy-to-learn machine-aided environment for processing. The offline text capture activity is designed to enable secretarial personnel to enter text effectively with minimal training and without transitional difficulties. The tutorial approach used in the online portion of MAE allows the novice user to accomplish his editorial goals with relative ease. But MAE is a developmental system for the exploration of a specific set of text production problems; it is not intended to be either general purpose or used directly in a production environment in its current implementation. It might be best characterized as a prototype of a production system, one that addresses specific application areas for which there are demonstrated needs but no current commercial offerings. 4d2

MAE can direct text to a variety of hard copy output devices. Text may be transferred to a cassette for playback at the IBM Selectric station using any of a selection of type spheres. Output may be produced on a medium-speed 96-character (upper/lower case) line printer. Output from MAE on magnetic tape can be directed to a commercial photocomposer to produce high-quality, camera ready masters. A MAE text file can be output on a medium acceptable to another external device for hard copy production (e.g., to cassette for transferral to the ARC utility for transmission over the ARPANET). 4d3

Documentation Development Community 4e

SRI is also creating a community of organizations interested in sharing long-range development of computer-based document production. The comminuty pools information, developments in procedures and softeware, and has access to the Augmented

Knowedge Workshop. The latter is a flexible, sophisticated compter infromation handling system suitable not only for document production but for development of prototype software and procedures, and for information exchange among participants. The products of all development work within the community are shared freely by all, but members also arrange separate, specialized work by SRI outside the community.* 4e1

*(Should client-proprietary work specific to NLS or MAE be allowed? Maybe no proprietary software, but yes proprietary procedures based on NLS or MAE? We don't need to talk about the question of possible proprietary work in NLS or MAE in this document, but we need to have a policy.) 4e1a

NLS provides the basis for flexible systems of creating, modifying, disseminating, and controlling documentation. NLS has particular advantages in easy modification of master copies, large-scale modification and reorganization of documents either as initial drafts or later for revision after publication, facil detailed editing, flexibility of printed output, including line drawings, and facile creation of special purpose subsystems. NLS is used as a medium to make printed or microfilm versions of files that are primarily intended for reading online an to publish material that would not otherwise be online. 4e1b

NLS has been used for over six years to produce, reports, small users' guides, proposals, and other technical documents for the Augmentation Research Center. Beginning in 1974 it has been used for publication in other organizations including Airforce documents in the range of 1000's of pages. 4e1c

4e2

Membership in the Documentation Development Laboratory [Or other Name] requires subscritpion to one slot in the Augmented Knowledge Workshop, payment of $N * for community membership and, at least half-time participation by a specialist in the employ of the subscriber. For subscriptions, member organizations receive the services associated with the AKW slot, the servies of an information exchange system, and a certain amount of consulting service from SRI specialists. In additon the community may sponser development work at SRI or by another member . The decision to allocate funds rests with the community membership and is executed by appropriate voting procedures. SRI will porpose development work and facilitate proposals from other members. We expect development work to cost community mebers of the order of $20,000 per year [?]. ** 4e3

*[Here is a proposal: For all such communities the community bank gets money by collecting $2000, of its own and $2000 from the slot for services, thus slots come to cost $42,000 of which ARC Applications gets $38,000] 4e3a

**[What happens when some member organization does not want to pay for develpoment work the majority desires? What happens when SRI does not want to do development work the majority desires? What control does SRI have over possible NLS development in other ogranizations so that it remains coherent with the rest of NLS?... We don't need to address the question of control of development work in this document but we need to have some sort of decision procedures written into contracts.] 4e3b

The Augmented Knowledge Workshop Slot 4e4

The Augmentation Research Center's Purpose in Supporting Communities of Users: 4e4a

The Augmentation Reserach Center (ARC) has existed for several years as a group researchers and system developers within SRI. The activities of ARC are aimed at exploring the possibilities for augmenting individuals and groups in the performance of knowledge work with the help of computer aids. 4e4a1

ARC's Initial Research and Development Strategy was for researchers within ARC do as much of their work as possible using the range of capabilities offered. Thus they have served not only as researchers, but also as the subjects for the analysis and evaluation of the augmentation systems for its first ten years. 4e4a2

The Next Stage in ARC's Research and Development Strategy has now begun. They are involving a wider group of people so that they can begin to transfer the fruits of their past work to others, and so that they can obtain feedback needed for further evolution from a wider spectrum of applications than occurs in the Center alone. Thes outsiders subscribe to ARC services as to an information utility. 4e4a3

ARPA and other government agencies have provided a considerable amount of funding for the development of the ARC technology during the past ten years. The Workshop Utility Service provides an effective medium for transfer of this technology to government, commercial, and educational organizations, thereby returning useful

results from the investment. As the community of using organizations grows, this return will become increasingly more significant. It is ARC's goal that these effects will be widespread in our society, both through direct use of the Workshop Utility and from use of related systems incorporating some aspects of the technology being developed here. 4e4a4

Establishing a Workshop Utility and providing the type of service discussed here are part of ARC's long-term commitment to pursue the continued development of Augmented Knowledge Workshops in a pragmatic, evolutionary manner. 4e4a5

The Workshop Utility Service 4e4b

The service includes: 4e4b1

Providing training as appropriate in the use of the ARC online system (NLS): Display NLS (DNLS), Typewriter NLS (TNLS), and Deferred Execution (DEX) software subsystems. 4e4b1a

Providing technical assistance to subscribing-organizations' "Workshop Architects" in the formulation, development, and implementation of augmented knowledge work procedures within their selected offices. 4e4b1b

This technical assistance includes help in the development of NLS use strategies suitable to each organization's environment, procedures within each organization for implementing these strategies, and possible special-application NLS extensions (or simplifications) to handle the mechanics of particular user needs and methodologies. 4e4b1c

The service also includes (and is based upon) the availability 20 hours a day, 7 days a week of Workshop Utility computer service, often via the ARPANET, from a PDP 10 TENEX system operated by a commercial facility management company, Tymshare, Inc. based in Cupertino, California. 4e4b2

Objective 4e4b3

The focus of our efforts is on working with subscribing organizations' personnel in the mutual development and use of procedures, methodology,

software features, and other online tools; and on the training of users that will allow their exploratory use of augmented Workshop systems 4e4b3a

Scope of the Workshop Utility Service 4e4b4

The types of Workshop services that we are beginning to support at varying levelsinclude: 4e4b4a

Collaborative Dialogue 4e4b4a1
Document Development, Production, And Control 4e4b4a2
Research Intelligence 4e4b4a3
Community Handbook Development 4e4b4a4
Computer-Based Instruction 4e4b4a5
Meetings And Conferences 4e4b4a6
Community Management And Organization 4e4b4a7
Special Knowledge Work By Individuals And Teams 4e4b4a8

Basic Contractual Arrangements: 4e4c

Clients are charged on the basis of direct and indirect services rendered. 4e4c1

The basic charge is $40,000 per year for each single "user-job-slot" access to the Office-1 computer at all times the system is currently scheduled to be available for use (20 hours/day, 7 days/week). This also covers direct and indirect people services. 4e4c1a

Information exchange consists of: [...an online newsleter, meetings...access to the journalized records of other publications efforts...personal efforts from SRI to facilitate information flow...etc. Needs to be spelled out further.] 4e5

Consulting services conists of: ...question answering from specialist in developments based on the NLS system, and related areas [We need to define this carefully] 4e6

Fourth DRAFT DDPCS Promotional Document

(J33121) 1-AUG-75 16:50;;;; Title: Author(s): Dirk H. Van Nouhuys/DVN; Distribution: /KLM([ACTION] docplan notebook please, and also, please, copies to Norm and Kaye Tomlin) DOCPLAN([INFO-ONLY]) ; Sub-Collections: SRI-ARC DOCPLAN; Clerk: DVN;

33121 Distribution
Kathey L. Mabrey, James H. Bair, David R. Brown, Glenn A. Sherwood, N. Dean Meyer, Kathey L. Mabrey, Norman R. Nielsen, Thomas L. Humphrey, Robert Louis Belleville, Elizabeth K. Michael, Richard W. Watson, James C. Norton, Robert N. Lieberman, Pat Whiting O'Keefe, Douglas C. Engelbart, Dirk H. Van Nouhuys,

DVN 1-AUG-75 17:07 33125

Status of Gathering Messages from Other Sites

Since I will be gone next week I thought I would report on the state of my effort to make a system to collect people's messages from more than one site. I plan a commands branch which will first collect messages from the message file at the home site, as it does now, then hand control over to a runfile via Run TENEX, which will log in through telent at a second site, ftp the message file there back to the home site where it will have a novel name, delete the message file at the foreign site, logout, and hand control back to the commands branch, which will treat the new file as it did the normal message file, and then repeat that cycle for however many foreign sites are involved. If it fails somewhere along the line, the message files it has already handled will appear on the printer. So far all I have done is make a first cut at the runfile (which did not work). It is in (office-1,vanNouhuys,tentacle,). 1

Status of Gathering Messages from Other Sites

(J33125) 1-AUG-75 17:07;;;; Title: Author(s): Dirk H. Van Nouhuys/DVN; Distribution: /DMB([INFO-ONLY]) SGR([INFO-ONLY]) KIRK([INFO-ONLY]) RWW([INFO-ONLY]) JCP([INFO-ONLY]) ; Sub-Collections: SRI-ARC; Clerk: DVN;

JGN 1-AUG-75 17:10 33126

Corrections to IDs

DS5 (Dana L> Small) is not authorized to use the NAVIMP directory. As such, both of her mail addresses should be NELC@ISIB(Attn: Dana). Please note the use of the routing subfield of the Net address. I do not know if NLS supports this feature, but if it doesn't, it should. If it is supported, please correct the network address of JGN (Greg Noel) to be NELC@ISIB(Attn: Greg). In addition, the delivery instructions for Dana should be Network (only). Thanks, Greg
P.S. Should her ID be DS5 or DLSn? 1

JGN 1-AUG-75 17:10 33126

Corrections to IDs

(J33126) 1-AUG-75 17:10;;;; Title: Author(s): J. Gregory Noel/JGN;
Distribution: /FEED([ACTION]) DSS([INFO-ONLY]) ;
Sub-Collections: NIC; Clerk: JGN;

FGB 1-AUG-75 19:36 33127

NUSC Directory

Jim & John,
This is to let you know that a NUSC directory has been established at Office-1. Your mail will be delivered there. You two are the only valid users; if you want others validated for use with this directory please let me know. Also, please acknowledge this message so I know that the Journal mechanisms are working properly.
Regards,

Frank
P.S. What the hexx is a N26 (NORD 26/Super Broussard) Do they have stewardesses & do they serve drinks? According to the airlines guide, that's what Allegheny flies into Groton.

1

FGB 1-AUG-75 19:36 33127

NUSC Directory

(J33127) 1-AUG-75 19:36;;;; Title: Author(s): Frank G. Brignoli/FGB; Distribution: /JPS([ACTION]) JCM3([ACTION]) ILA([INFO-ONLY]) ; Sub-Collections: NIC; Clerk: FGB;

JI JE 2-AUG-75 10:16 33128

A COLLABORATION SUPPORT SYSTEM

An Online, Network-Oriented Working Environment

By
Jean Iseli
James E, Eagle
MITRE Corporation, McLean, Virginia

12 March 1975

The past several years have seen increasing interest and activity in the development of large scale intercomputer networks based on the technology represented by the ARPANet. These networks are viewed by their potential importers as powerful means for supporting the fulfilment of their operational charters. There is, however, a crucial difference between the goals of the ARPANet and the goals of those who would seek to use the enabling technology to their own ends. 2

The ARPANet effort has been successful in developing a unified technology of widespread potential benefit. In a sense it has been a theory proving exercise and it has worked. 3

The ARPANet, however, is not burdened by an operational, mission-oriented responsibility - and rightly so. It has no cohesive body of users who can collectively be viewed as the consumer of the products of network operation. The varied requirements of users not directly concerned with network development have raised more questions about general network theory than they have about specific end-user satisfaction. In the ARPANet atmosphere this is probably a proper ordering of priorities. Indeed, it may be said in fairness and without criticism that the ARPANet has approached the state of a tool whose major useful activity is the honing of itself. The goal of the ARPANet is a better ARPANet whose goal is a still better .....
4

The prospective importer of this body of network knowledge is understandably more concerned with its external usefulness than its internal elegance. Unfortunately, admiration of the latter leads to a sense of complacency about the achievability of the former. 5

Recent experience has demonstrated that attempts to transplant this hothouse technology into a climate of focused utility in behalf of a concerned proprietor have quickly encountered an interesting but uncomfortable fact: the technology of network construction has far outstripped the technology of potential network utilization. 6

The potential of networks to allow distributed users to collaborate on the achievement of shared objectives has been well described <1,2>. Moreover, numerous capabilities are or have evolved to facilitate such joint working <3,4,5,6,7,8,9,10,11,12>. Unfortunately, the richness of such capabilities is still largely denied to all but the honers of the technology. 7

Reasons for this denial include: 8

(a) The user is still required to interact through media and interface languages unfamiliar and unnatural to him and their mastery often requires more investment than warranted on the basis or rendered utility. 8a

(b) Network resources, although rich in capability, are still individual, often incompatible, and practically inaccessible to all but sophisticates because of required familiarity of multiple hardware and software systems and the investment required to achieve such familiarity. 8b

(c) Use of multiple resources for distributed achievement of shared objectives requires significant expenditure of energy in non-mission oriented pursuits to locate, access, understand, use desired resources, and ameliorate network difficulties. 8c

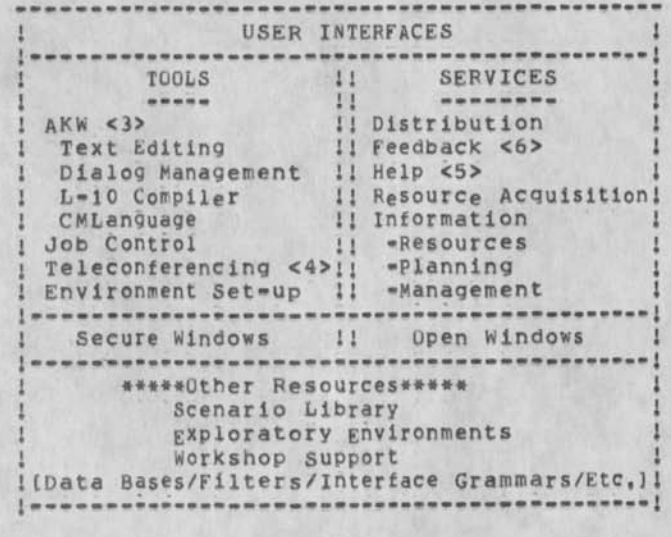
(d) Users have no network access to detailed information about the network, its resources and their use. Existing such access requires extensive pre-knowledge which is difficult to obtain. 8d

(e) Few mechanisms exist to enable responsive on-line dialog between users, planners, designers, implementors, maintainers, and operators to ensure a convergence between capability evolution and user requirements. 8e

The solution to these problems may be approached through the development of a mechanism that: (1) provides common connective tissue between individual resources and capabilities, (2) allows the user to work through media and language consistent with his natural expertise, and (3) allows the user to compatibly enlarge and augment his environment as his knowledge and expertise grows. Such a mechanism is described herein as a Collaboration Support System. 9

This mechanism, the Collaboration Support System, provides a user environment which combines the best individual computer resources and capabilities presently available in such a fashion that each increment of user investment yields a direct, reliable and replicatable measure of utility. It is particularly useful to users who are geographically separated and organizationally disparate, allowing them to share knowledge, information, resources, and dialog in the achievement of shared and/or negotiated objectives. 10

To simplify discussion of the CSS, the following pictorial representation is provided to focus the discussion:



11

Logical components 11a

The major logical components of the CSS as they are seen by a user are illustrated above. The specific composition of interfaces, tools, services, windows and other resources for any particular instance of CSS usage are specifiable by a user -- either by designating a previously specified environment by name, or through his selection of his environment (through the environment set-up capability) from a menu of possible environment components. These consist of: 11a1

User Interfaces -- CSS design allows for different interfaces to each community of users; in fact, each CSS user may specify his own interface. A mechansim that may be employed to generate this capability is the CML <12> -- there would be a translation process between the user interface and the "worker software" that implements the commands. The specifics of any user interface would be stored within the CSS as an "interface grammar". The CSS also contains "filters" [specialized user interfaces] to remote tools and services. These filters are required to simplify user interfaces consistently and to provide simple resource acquisition services. 11a1a

Tools -- CSS design allows tools to be either resident within the CSS or available to users from "tool bearing" <8> hosts on a relevant network. It is assumed that tools are available to users either as they exist on the tool bearing host, or, that they have been "front-ended" by a "filter" such that the command interface to the user is better adapted to his needs. Not all tools will be filtered; in fact, the filter library is envisioned to be an evolving one and will represent a major area of exploratory development. Conceptually, CSS design allows for the functional stratification of tools; that is, tools or portions of tools will be restructurable to exist either in the CSS proper or within "tool bearing" host environments. This type of architecture is already being implemented within the proposed NSW and the NLS front- back- ends. The prior figure illustrates some of the tools that will be available within the CSS. 11a1b

Some typical tools that will be provided, locally and/or remotely, include: teleconferencing, job control [for the creation, control and use of multiple asynchronous jobs from one user terminal], and environment control [to provide for process scheduling, event monitoring-notification-action, and intelligent communications buffering]. 11a1c

Services -- Within any user environment, the abilities to: distribute entities [data, information, programs] to other persons; engage in two-way dialog with planners, designers, implementors, maintainers; secure static and dynamic assistance (help) in the utilization of resources; acquire resources; and acquire information, are essential. Within the CSS, these are referred to as services which are both CSS resident and available as distributed entities on interconnecting networks through appropriate interface mechanisms [windows]. 11a1d

Windows -- Since the CSS is an environment for distributed persons to collaborate as well as a user specific environment, windows to networks will be provided. Such windows will be both open and secure; and dynamically connectable and disconnectable. The windows will provide for access to resources on other existing networks, either directly or through CSS filtering. 11a1e

Other Resources -- In addition to the above, the CSS will contain other resources such as: a scenario library (for demonstrations, user-training and exercises), an environment library (containing user-specific environment set-up specifications as well as typical [often used] environment set-ups), assorted workshop support facilities that will evolve as CSS utilization accrues, interface grammars, filters, specialized data bases, compilers, and other development support capabilities. 11a1f

Architecture 11b

As presently envisioned, the CSS will be realized in a mini-computer the size of a PDP-11/45 to serve also as a prototypical intelligent terminal - it being assumed that comparable computational capability will be available within the next five year time frame in a user terminal. 11b1

A major feature of the CSS will be that it can be dynamically functionally endowed from a network endowment center. This implies that the CSS architecture employs strict implementation disciplines including: 11b2

Functional Modularity - wherein each functional module contains internal user documentation and assistance [ascribing to a CSS construct protocol], and is amenable to hierarchical aggregation. 11b2a

Structured Implementation - providing inter-module communications and uniform control transition mechanisms. 11b2b

Functional Extensibility - providing for functional augmentation at any hierarchical level. 11b2c

Coupled with user specific interface definition capability, the above features will allow CSS to be dynamically functionally endowed through a network endowment center in response to the unique needs of specific user communities. Further, this form of architecture provides for new alternatives with respect to security and privacy. 11b3

Summary 12

The CSS is a step towards providing the specific tools and services required by mission and/or discipline oriented communities in such a fashion that each increment of user investment will lead to a correspondingly larger increment of derived utility. It is also a step in the direction of establishing a prototypical intelligent terminal that may be employed operationally to assist in the evolution of intelligent terminal design. Its construct recognizes that computer power will migrate towards a local user capability and networking will be the mechanism to ensure inter-user communications and access to specialized resources which cannot be vested in user terminals; that the predominant expense will continue to be software construction - thus supporting the functional endowment approach; and that users require the ability to easily assemble for their use those portions of the technology that can be of benefit to them. 12a

It is hoped that this description will stimulate dialog towards new approaches for decreasing the distance between where technology is and its ability to render benefit to an increasingly diverse and varied user population. 12b

REFERENCES 13

<1> Douglas C. Engelbart. Intellectual Implications of Multi-Access Computer Networks. Paper presented at Interdisciplinary Conference on Multi-Access Computer Networks, Austin, Texas, April 1970. Preprint, 12p. NIC 5255. 13a

<2> D. C. Engelbart. Coordinated Information Services for a Discipline- or Mission-Oriented Community. Stanford Research Institute, Augmentation Research Center, Menlo Park, California. Paper given at Second Annual Computer Communications Conference, San Jose, California, 24 January 1973. 12 December 1972. Preprint, 13p. NIC 12445. 13b

<3> D. C. Engelbart, R. W. Watson, J. C. Norton. The Augmented Knowledge Workshop. 1 March 1973. 19p. NIC 14724. 13c

<4> J. Iseli, et al. Capability Description of a Digital Teleconferencer. 26 December 1974. 27p. MITRE-WP-10813. 13d

<5> J. Iseli, H. Sternick, S. Poh. Description of a Proposed ARPANet Help Facility. July 1974. 33p. MITRE-MTR-6723. 13e

<6> J. Calvin, D. Crocker, J. Iseli, and A. Rosenfeld. Recommendations for a Network User Feedback System. Using Note #10, NIC 21683. 13f

<7> Consistent System 13g

<8> National Software Works [NSW] 13h

Jonathan B. Postel, "National Software Works Protocols - Version 2," 1, January 1975, Augmentation Research Center, Stanford Research Institute, Accession NIC 24856. 13h1

James E. White, "The Procedure Call Protocol - Version 2," 1 January 1975, Augmentation Research Center, Stanford Research Institute, Accession NIC 24855. 13h2

<9> ARPANet Protocols 13i

<10> J. Vallee, H. M. Lipinski, R. H. Miller. Group Communications Through Computers. July 1974. R-32. Institute For the Future. Menlo Park, California. 13j

<11> Calvin, James O, "The Design and Implementation of an Interactive Teleconferencing Environment," 10 May 1974, [Submitted in partial fulfillment for the requirements for the Bachelor of Science Degree in Undergraduate Scholars], Case Western Reserve University. 13k

<12> NLS - References to Augmentation Research Center Journal System in terms of Accession numbers: 131

"Syntax of NLS Commands: New Command Language," 4, May 1974. 1311

"TNLS-8 Equivalents of NLS-7 Commands," 16 October 1974, NIC 23913. 1312

"Basic TNLS Course," 30 May 1974, NIC 22858. 1313

"L10 Users" Guide," 11 September 1973, NIC 18969. 1314

"Updated CML Documentation," 21 November 1973, NIC 20438. 1315

"Output Processor Users" Guide," 23 August 1973, NIC 12209. 1316

JI JE 2-AUG-75 10:16 33128

(J33128) 2-AUG-75 10:16;;;; Title: Author(s): Jean Iseli, James Eagle/JI JE; Distribution: /DCE([ACTION]) RWW([ACTION]) JCN([ACTION]) CHI([ACTION]) JBP([ACTION]) JAKE([ACTION]) TEH([ACTION]) JNH([ACTION]) MEJ([ACTION]) SGG([ACTION]) THP([ACTION]) JEG([ACTION]) JT4([ACTION]) STW([ACTION]) KM([ACTION]) HS2([ACTION]) JE([ACTION]) MEJ([INFO-ONLY]) ; Sub-Collections: NIC; Clerk: MEJ; Origin: < HELP, COMPCON,NLS;14, >, 12-MAR-75 21:01 JI ;;;; ####;

test of journal system

xx 1

if this data is in the journal copy theb we know that the pc gets merged in. 2

test of journal system

(J33131) 2-AUG-75 12:45;;;; Title: Author(s): William E. Carlson/WEC; Distribution: /WEC([ACTION]) ; Sub-Collections: NIC; Clerk: WEC; Origin: < CARLSON, TESTCASE,NLS;2, >, 2-AUG-75 12:44 WEC ;;;;####;

Archived Copy of Steering Committee Guidance to COMPASS on 3 July 75

WEC 2-AUG-75 14:53 33132

Archived Copy of Steering Committee Guidance to COMPASS on 3 July 75

Mail from USC-ISI rcvd at 3-JUL-75 1455-PDT
Date: 3 JUL 1975 1436-PDT
From: CARLSON at USC-ISI
Subject: Steering Committee Guidance
To: warshall at BBNB
cc: [BBNB]<POSTEL>NSW-DISTRIBUTION.LIST; 2

Dear Steve: 3

The Steering Committee would like to thank you for taking on the task of technical coordination for NSW System Software. Attached is a statement of the Steering Committee's objectives during the next eighteen months, NSW contractors will look to COMPASS for resolution of technical issues and priorities for their work. The Steering Committee will support you in every way, including allocation of additional funds and changes to Statements of Work, so that these objectives can be attained on schedule. 3a

ARPA - Bill Carlson
AFDAA - L/C Tony Baggiano
AFDSDC - 1Lt Larry Crain
AFDSC - Maj Gary Hignett
RADC - Capt Mike Wingfield
RADC - Duane Stone 3a1

attch:
********** 4

ARPA and the Air Force are jointly developing the National Software Works (NSW) to demonstrate a new approach to reducing both the cost and the time for DOD agencies to design, develop, test, document, and maintain computer software. A

great deal of software to aid in program design, implementation, etc, has been developed, but numerous problems have been encountered in the distribution of these aids. A mechanism is needed to make these existing software tools available in a uniform manner to DoD. Thus, the concern of NSW is the development of this delivery mechanism (rather than the development of the tools themselves). This focus on the delivery mechanism is also required for the implementation of automated access control and management strategies, a necessary complement to the software tools. The consequences of this approach can be characterized as follows: 4a

- Software development and maintenance will take place in the software world's equivalent of a factory. The factory is to contain "tools" which aid programmers and managers. Despite the term "factory", no buildings are to be constructed, and for the most part existing computer facilities can be used. 4a1

- A large fraction of the software development and maintenance tools will be machine independent. That does not mean that tools (which are themselves computer programs) can run on arbitrarily chosen computers. Machine independence is meant to imply that computer specialists building software for many different kinds of computers can share a single version of a tool. Examples of capabilities which can be shared include text editors, the filing system, bookkeeping operations, project management tools, program test data generators, and documentation aids. 4a2

- Compilers and other execution dependent development and maintenance tools will be provided separately for each target machine (machine on which the software which is being written will run). 4a3

- The tools will be embedded in a resource-sharing framework which minimizes the user's awareness of the fact that resources are scattered over many hosts. Thus, the programmer requests resources in a uniform way, without having to learn the idiosyncrasies of different host operating systems or log-in procedures. The manager can contract for services with the NSW, without having to negotiate accounts on each host. 4a4

- A central design criterion for the framework is that it minimize the difficulties of appending new tools and new resource-bearing hosts to the set of facilities in the NSW. 4a5

- The framework will provide the project manager with much tighter controls over his project than are currently in use. There will be unified and consistent mechanisms for access and accounting control of resources, where the resources include tools and files on several dissimilar hosts. The control mechanisms will support sophisticated tools for the specification and implementation of management policies of project control. These policies typically will specify who may access and/or change which modules, what crosschecks must be carried out whenever a module is changed, and what consequent actions are required for each change. 4a6

Planning for the NSW began in the Summer of 1973, with development starting a year later in July 1974. The initial version of the National Software Works is being implemented on the ARPANET. As capabilities become available, they will be tested by the Air Force Data Systems Design Center at Gunter AFB, Alabama and by the Air Force Data Services Center in the Pentagon. Software for the Burroughs B3500, which is the standard Air Force Base level computer, is developed at Gunter

and distributed to over 130 Air Force installations throughout the world. An important goal is to demonstrate the NSW's effectiveness in supporting the development of software to run on the B3500. 4b

The heart of the NSW system is a program called the Works Manager, which is a family of procedures designed to provide a unified and consistent mechanism for access control and accounting. The Works Manager has as its central data base a catalogue of resources. The information this catalogue contains about tools, users, and files is believed to be sufficiently general to represent extremely sophisticated strategies of management control and accounting, involving quite complex rules of access control and sophisticated strategies of fund commitment and expenditure restriction. 4c

Users will access the NSW through a hardware/software system called the Front-End, which provides a consistent interface to software development and maintenance tools running on a variety of dissimilar computers. The Front-End will standardize the control characters the user must type to interrupt an activity, initiate a command, erase a character or word, etc. The Front-End will also allow a wide variety of different kinds of terminals to be used with the NSW. Device dependent code will be isolated in the Front-End, so the remainder of the NSW can assume that terminals obey the conventions of a small number of device classes. 4d

It has been hypothesized that the Front-End can control almost all user interactions with NSW tools, using tables which define tool command languages. The NSW can operate with a less intelligent Front-End, but tighter standards for interactive tool user interfaces can be enforced if they share a common command interpreter. A prototype Front-End which

offers generalized command interpretation is being constructed, and the production version of the Front-End will offer this service if the prototype is found to be sufficiently robust to support most if not all manners of dialogue between users and tools, and if the operating costs are not excessive, 4e

In order for the Works Manager to control resources in the collection of dissimilar computers where software tools will run, a set of standard software functions must be implemented in each of those computers. The functions will be invoked by the Works Manager using standard communications protocols. Functional requirements for this software to be implemented in each Tool-Bearing Host have been defined, and a first attempt at specifying the software and communications protocols in detail has been made. Initial Tool-Bearing Hosts will be TENEX, MULTICS, an IBM 360/91, and a Burroughs B4700 connected through a Network Connection Subsystem. As these implementations proceed, the specifications will be reviewed and revised in ways which simplify implementation but still satisfy the functional requirements for the Works Manager to control resources throughout the NSW. 4f

During the eighteen months, July 1975 to December 1976, the initial NSW system will be completed and evaluated, and the feasibility of expanding the system and operating it cost-effectively for production applications will be determined. The experiments will provide the NSW Steering Committee with the information it needs to project usage requirements. An analysis of the cost/performance tradeoffs in building a scaled, hardened, optimized version of the NSW will be conducted. A detailed design will be prepared for a system which represents a reasonable compromise between

performance and cost in satisfying projected usage requirements. 4g

Archived Copy of Steering Committee Guidance to COMPASS on 3 July 75

(J33132) 2-AUG-75 14:53;;;; Title: Author(s): William E. Carlson/WEC; Distribution: /WEC([ACTION]) LAC([INFO-ONLY]) MAW([INFO-ONLY]) CSH([INFO-ONLY]) ; Sub-Collections: NIC; Clerk: WEC; Origin: < CARLSON, NSW-OBJECTIVES,NLS;6, >, 2-AUG-75 14:46 WEC ;;;;####;

JHB 3-AUG-75 13:02 33134

Request for Hardware Support for NLS Public Seminar

Martin, It looks like what we are going to need will be as follows (by Aug 15 at the latest, earlier would give us more time to work bugs and procedures out): 1

3 Data Media workstations hooked up to our fastest ELF and located along the East wall of our conference room 1a

Removal of the TV equipment to provide room and a less cluttered appearance 1b

A tuned up, repaired, ready to go Video Projector, hooked up to one of the Data Medias. 1c

The Terminet hooked up as a printer through one of the LPs to demonstrate that capablity you developed and provide an intermediate printing capability. 1d

An option for a fourth workstation if more people sign up. 1e

(Some touch up painting ... any suggestions how we could get this to happen?) 1f

JHB 3-AUG-75 13:02 33134

Request for Hardware Support for NLS Public Seminar

(J33134) 3-AUG-75 13:02;;;; Title: Author(s): James H. Bair/JHB;
Distribution: /MEH([ACTION]) JCN([INFO-ONLY]) SGR([INFO-ONLY]
) DCE([INFO-ONLY]) RLL([INFO-ONLY]) BJP([INFO-ONLY]) ;
Sub-Collections: SRI-ARC; Clerk: JHB;

JHB 3-AUG-75 13:23 33135

Request for directories to be used by particpants during Public NLS Seminars

Bud, Please have directories set up as follows (will need by Aug 15th): 1

6 directories: SEMINAR1 SEMINAR2 SEMINAR3 SEMINAR4 SEMINAR5 SEMINAR6 1a

passwords = same as directory name 1a1

account number = whatever you determine 1a2

disc pp = 100 ea, 1a3

allocation group = ARC Applications, unless you think they should be in special group (most use will be after 2pm PDT though) -- will need Minimum 1.5 % system per workstation 1a4

(May need more directories if more people sign up) 1a5

JHB 3-AUG-75 13:23 33135

Request for directories to be used by particpants during Public NLS Seminars

(J33135) 3-AUG-75 13:23;;;; Title: Author(s): James H. Bair/JHB;
Distribution: /BJP([ACTION]) JCN([INFO-ONLY]) DCE([INFO-ONLY]
) SGR([INFO-ONLY]) RLL([INFO-ONLY]) JCP([INFO-ONLY]) ;
Sub-Collections: SRI-ARC; Clerk: JHB;

RA3Y 3-AUG-75 16:54 33136

A Party

I will be having a party this Friday (August 8), to celebrate the completion of my dissertation. The party begins at 8:00 p.m.. My address is 1950 Cooley Ave, Apt 5212, in East Palo Alto. If you can come, please let me know so I can make preparations. Hope to see you there. 1

A Party

(J33136) 3-AUG-75 16:54;;;; Title: Author(s): Raymond R. Panko/RA3Y; Distribution: /JCN([ACTION]) JHB([ACTION]) RLL([ACTION]) DCE([ACTION]) BJP([ACTION]) GCE([ACTION]) RWW([ACTION]) JLE([ACTION]) ; Sub-Collections: SRI-ARC; Clerk: RA3Y;

DLS 4-AUG-75 12:36 33139

Merge source code and index (as of 25 JUL 75)

for the record

Merge source code and index (as of 25 JUL 75)

```
FILE merge                                                          1
  DECLARE TEXT POINTER sf, sl, pt1, pt2;                            1a
  DECLARE STRING word1[100], word2[100];                            1b
  (merge) PROCEDURE;                                                1c
  % process index branch %                                          1d
    FIND "sf;                                                       1d1
    IF NOT getft1(sf) THEN                                          1d2
      BEGIN                                                         1d2a
      LOOP                                                          1d2b
        BEGIN                                                       1d2b1
        sl _ getsuc(sf) ;                                           1d2b2
        FIND SF(sf) "pt1 [SP] "pt2 ;                                1d2b3
        *word1* _ pt1 pt2 ;                                         1d2b4
        FIND SF(sl) "pt1 [Sp] "pt2 ;                                1d2b5
        *word2* _ Pt1 Pt2 ;                                         1d2b6
        IF *word1* = *word2* THEN                                   1d2b7
          BEGIN                                                     1d2b7a
          FIND SE(sf) "pt2 $PT "pt1 > ;                             1d2b7b
          *word1* _ pt1 pt2 ;                                       1d2b7c
          FIND SE(sl) "pt2 $pT "pt1 > ;                             1d2b7d
          *word2* _ Pt1 Pt2 ;                                       1d2b7e
          IF *word1* = *word2*                                      1d2b7f
            THEN                                                    1d2b7f1
              cdelsta (sl)                                          1d2b7f1a
            ELSE                                                    1d2b7f2
```

```
                    BEGIN                                                1d2b7f2a
                    ST sf _ SF(sf) SE(sf), ", ", pt1 pt2 ;               1d2b7f2b
                    cdelsta (sl) ;                                       1d2b7f2c
                    END;                                                 1d2b7f2d
                IF getftl(sf) THEN EXIT LOOP ELSE REPEAT LOOP ;          1d2b7g
                END;                                                     1d2b7h
            IF getftl(sl) THEN EXIT LOOP ;                               1d2b8
            END;                                                         1d2b9
          RETURN(FALSE);                                                 1d2c
          END;                                                           1d2d
        END,                                                             1d3
        FINISH                                                           1d4
Index to group from 2 to 217                                             2
    !ABNORMAL  <2>                                                       2a
    !BEGIN  <17>                                                         2b
    !COMPOOL  <35>                                                       2c
    !COPY  <47>                                                          2d
    !END  <65>                                                           2e
    !FREQUENCY  <81>                                                     2f
    !INTERFERENCE  <93>                                                  2g
    !LINKAGE  <102>                                                      2h
    !ORDER  <122>                                                        2i
    !POINTER  <136>                                                      2j
    !RECURSIVE  <147>                                                    2k
    !SETS  <153>                                                         2l
```

DLS 4-AUG-75 12:36 33139

Merge source code and index (as of 25 JUL 75)

(J33139) 4-AUG-75 12:36;;;; Title: Author(s): Duane L. Stone/DLS;
Distribution: /WK4([INFO-ONLY]) RN2([INFO-ONLY]) ;
Sub-Collections: RADC; Clerk: DLS;

33139 Distribution
Wolf-Hasso Kaubisch, Richard Nelson,

JML 4-AUG-75 14:08 33140

JML's Weekly Report for Week of 7/28

This is just to let you know what I'm doing. I tend to work on the regular schedule of 9 to 6, which doesn't seem to be causing any dislocations. It works out perfectly in my home-and-people schedule.

JML 4-AUG-75 14:08 33140

JML's Weekly Report for Week of 7/28

Above and beyond my usual printout-ripping and stapling, text inputting, and XDUCing tasks, I worked on the following this week: 1

I continued work on shaping up the Final Report -- putting in the edits of RWW and DCE, and the sections which the authors rewrote, I kept my eyes open for all anomalies in spelling, consistency of chapter organization, grammatical unconstructs, incomplete references, etc., and cleared them up whenever possible. 1a

I coached Maryann on Central Office procedures. 1b

I interviewed Jeff Peters for information on BSYS, which needed a short writeup in the Final Report. I will write up my notes and submit them for approval to RWW before inserting -- this is on the order of one or two paragraphs which RWW feels very strongly were sorely missed. 1c

As the only person present at ARC able to do so this week, I taught DEX to the trainers Pam and Priscilla. The DEX effort is moving into fuller swing, and DEX is going to be taught at Gunter the week of 8/4 by Susan and Priscilla. As part of this effort Jeanne Beck et al were rewriting the DEX Primer for the ICP Termicette, and through the week I answered questions about DEX as they arose. 1d

I continued to stay on top of the onslaught of Journal hardcopy for the master and access notebooks. I intend to send out a journal item(s) informing people that these hardcopy sources are up-to-date and can be referred to. 1e

To Do Next Week 2

I intend to spend the larger part of my time on the Fiinal Report, trying to complete it substantively by the end of the week. 2a

JML's Weekly Report for Week of 7/28

(J33140) 4-AUG-75 14:08;;;; Title: Author(s): Jeanne M. Leavitt/JML;
Distribution: /DCE([INFO-ONLY]) JCN([INFO-ONLY]) ;
Sub-Collections: SRI-ARC; Clerk: JML; Origin: < LEAVITT,
WEEK7/28.NLS;2, >, 4-AUG-75 13:57 JML ;;;;####;