LAC 17-JUN-75 20:40 32766

NSW Plan, parts I,II -see comment

mike, heres the first two parts of the plan, parts 3-5 got fritzed somehow, i guess in the crash they had at SRI. Please check these over carefully, and I'll get you the remainder when (IF) i can get it recovered. /Larry

1 PART I: OVERVIEW OF THE NATIONAL SOFTWARE WORKS PROJECT 1

1a Introduction 1a

1a1 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. 1a1

1a2 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. 1a2

1a3 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. 1a3

1b Background and Technical Need 1b

1b1 The Software Problem (Excerpted from Ref. 2) 1b1

1b1a 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. 1b1a

1b1b 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. 1b1b

1b1c 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. 1b1c

1b1d 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. 1b1d

1b1e 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. 1b1e

1b1f 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. 1b1f

1b1g 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. 1b1g

1b2 Why So Few Tools? 1b2

1b2a 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. 1b2a

1b2b 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. 1b2b

1b2c 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. 1b2c

1b2d 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. 1b2d

1b2e 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. 1b2e

1b3 Origins of the National Software Works Program 1b3

1b3a 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. 1b3a

1b3b 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. 1b3b

1b3c 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. 1b3c

1b3d 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. 1b3d

1b3e 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.) 1b3e

1b3f 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). 1b3f

1b3g 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. 1b3g

1b3h 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. 1b3h

1c National Software Works Design Concepts 1c

1c1 Overview 1c1

1c1a 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. 1c1a

1c1b 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. 1c1b

1c1c 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 10S0 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. 1c1c

1c1d 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. 1c1d

1c1e 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. 1c1e

1c1f 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. 1c1f

1c1g 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. 1c1g

1c2 A New Capability for Project Control 1c2

1c2a 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. 1c2a

1c2b 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. 1c2b

1c2c 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. 1c2c

1c2d 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). 1c2d

1c2e 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). 1c2e

1c2f 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. 1c2f

1c2g 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. 1c2g

1c2h 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 provice 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. 1c2h

1c2i 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: 1c2i

1c2i1 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. 1c2i1

1c2i2 Project Accountant: This produces reports on the frequency and cost of various patterns of activity interesting to project management. 1c2i2

1c2i3 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. 1c2i3

1c2j 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. 1c2j

1c3 Internal Design 1c3

1c3a 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). 1c3a

1c3a1 (Referencexxx) 1c3a1

1c3b Works Manager 1c3b

1c3b1 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. 1c3b1

1c3b2 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. 1c3b2

1c3b3 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). 1c3b3

1c3c Foreman 1c3c

1c3c1 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. 1c3c1

1c3c2 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. 1c3c2

1c3d Front End 1c3d

1c3d1 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. 1c3d1

1c3d2 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. 1c3d2

1c3d3 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. 1c3d3

1c3d4 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. 1c3d4

1c3d5 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. 1c3d5

1c3d6 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. 1c3d6

1c4 Generalized Support tools 1c4

1c4a Yellow pages 1c4a

1c4a1 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. 1c4a1

1c4b Software libraries. 1c4b

1c4b1 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. 1c4b1

1c4b2 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). 1c4b2

1c4b3 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: 1c4b3

1c4b3a 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. 1c4b3a

1c4b3b The subroutine can be called across the network for debugging, and the user would again be charged a stiff exit fee. 1c4b3b

1c4b3c Only certified linkage tools are allowed to access the subroutine library, and the user is charged each time he builds a new object module. 1c4b3c

1c4b4 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. 1c4b4

1c4b5 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. 1c4b5

1c4c Batch scheduler. 1c4c

1c4c1 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. 1c4c1

1c4c2 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. 1c4c2

1c4c3 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). 1c4c3

1c4c4 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. 1c4c4

1c4d Software conversion aids 1c4d

1c4d1 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. 1c4d1

1c4d2 The NSW environment will help this problem in three ways: 1c4d2

1c4d2a 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. 1c4d2a

1c4d2b The NSW will include a collection of tools for converting production software to run on different (and normally newer) hardware. 1c4d2b

1c4d2c 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. 1c4d2c

1c5 Management Considerations 1c5

1c5a Usage Audit Trail 1c5a

1c5a1 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. 1c5a1

1c5b Maintenance Categories 1c5b

1c5b1 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. 1c5b1

1c5c Cost Predictions and Recovery 1c5c

1c5c1 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. 1c5c1

1c5c2 Efficient use of NSW facilities must be encouraged by charging and reimbursement policies which reflect the true economic value of the services used. 1c5c2

1d Implementaton Guidelines 1d

1d1 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. 1d1

1d2 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. 1d2

1e Participating Organizations 1e

1e1 The NSW Steering Committee is currently composed of representatives of the following government agencies which are supporting the project: 1e1

1e1a Air Force Data Automation Agency(AFDAA) 1e1a

1e1a1 Air Force Data Services Center 1e1a1

1e1a2 Air Force Data Systems Design Center 1e1a2

1e1b Rome Air Development Center 1e1b

1e1c Defense Advanced Research Projects Agency(ARPA) 1e1c

1e2 The following Commercial organizations and universities are under contract to assist in the development of various portions of the system: 1e2

1e2a Applied Data Research(ADR) 1e2a

1e2b Massachusetts Computer Associates (COMPASS) 1e2b

1e2c Stanford Research Institute(SRI) 1e2c

1e2d Bolt Beranek and Newman(BBN) 1e2d

1e2e MIT Project MAC 1e2e

1e2f UCLA Campus Computing Network(CCN) 1e2f

1e2g Speech Communications Research Laboratory(SCRL) 1e2g

1e2h Computer Corporation of America(CCA) 1e2h

1e2i Science Applications, Inc.(SAI) 1e2i

1e3 The NSW Advisory Committee is composed of representatives of the following Government agencies interested in the project: 1e3

1e3a Army Material Command 1e3a

1e3b Headquarters, A.F. Systems Command 1e3b

1e3c *****need a list***** 1e3c

2 PART II: PERSPECTIVE OF PARTICIPATING DOD ORGANIZATIONS' VIEW OF THE NSW 2

2a Air Force Data Services Center 2a

2a1 The AFDSC Environment 2a1

2a1a 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. 2a1a

2a1b 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. 2a1b

2a1c 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. 2a1c

2a1d 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. 2a1d

2a1d1 TABLE 1: AFDSC Resources Devoted to Software Development 2a1d1

2a1d1a	FY73	FY74	FY75	
2a1e Non-Multics Development vs Total Manhrs	19.9%	19.5%	15.5%	2a1e
2a1f Multics Dev Total Manhrs	0	3.6%	4.2%	2a1f
2a1g All Dev vs Total Manhrs	19.9%	23.1%	19.7%	2a1g
2a1h				2a1h
2a1i Computer Time For Dev vs All Computer Time (360/75 and GCOS only)	7.5%	7.7%	4.8%	2a1i

2a1j 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. 2a1j

2a1k 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. 2a1k

2a2 AFDSC Operated Computer Service Bureaus 2a2

2a2a 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. 2a2a

2a3 AFDSC Requirements For NSW 2a3

2a3a Usage Scenarios 2a3a

2a3a1 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. 2a3a1

2a3b Tool Requirements, by Target Machine 2a3b

2a3b1 Honeywell G635 or H6000 with GCOS 2a3b1

2a3b1a The primary requirement is for testing batch jobs . Source level interactive debugging is required for COBOL and FORTRAN. 2a3b1a

2a3b1b 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. 2a3b1b

2a3b1c Tools are required for designing, implementing, and restructuring random, ISP, and IDS data bases. 2a3b1c

2a3b1d 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). 2a3b1d

2a3b1e 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. 2a3b1e

2a3b2 IBM 360/370 With OS 2a3b2

2a3b2a There are requirements for FORTRAN, COBOL, and possibly PL/1 development on the 360/370. For each language, source level interactive debugging is required. 2a3b2a

2a3b2b 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. 2a3b2b

2a3b2c 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. 2a3b2c

2a3b2d 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. 2a3b2d

2a3b2e 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. 2a3b2e

2a3b2f 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 IBH, but be used to build software for, as an example, a Honeywell 6000 with GCOS. 2a3b2f

2a3b3 Multics 2a3b3

2a3b3a 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. 2a3b3a

2a3c Requirements For Tools For Controlling And Maintaining Existing Software Systems. 2a3c

2a3c1 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 establisned 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. 2a3c1

2b Air Force Data Systems Design Center 2b

2b1 THE AFDSDC ENVIRONMENT 2b1

2b1a AFDSDC Mission 2b1a

2b1a1 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. 2b1a1

2b1a1a The only exception to this mission assignment is the personnel function which is the responsibility of the Military Personnel Center. 2b1a1a

2b1a1b 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. 2b1a1b

2b1a2 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. 2b1a2

2b1a3 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. 2b1a3

2b1b Organization 2b1b

2b1b1 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. 2b1b1

2b1b1a 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. 2b1b1a

2b1b1b 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. 2b1b1b

2b1b2 A relatively large Auditor staff is co-located with the Center to assure adequacy of audit trails and management during system development. 2b1b2

2b1b3 A Communications detachment from the Air Force Communications Service is also co-located with the Center to provide ongoing communications expertise. 2b1b3

2b1b4 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. 2b1b4

2b1c Responsibilities, Resources, and Workload 2b1c

2b1c1 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 . 2b1c1

2b1c2 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. 2b1c2

2b1c3 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. 2b1c3

2b1c4 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. 2b1c4

2b1c5 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. 2b1c5

2b2 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. 2b2

2b2a 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. 2b2a

2b2b 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. 2b2b

2b2c 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). 2b2c

2b2d 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. 2b2d

2b3 SCENARIOS OF OPERATION: CURRENT AND WITH NSW 2b3

2b3a B3500/B4700 Software Production
---------------------------------- 2b3a

2b3a1 CURRENT ENVIRONMENT 2b3a1

2b3a1a 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. 2b3a1a

2b3a1b 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. 2b3a1b

2b3a2 NSW ENVIRONMENT 2b3a2

2b3a2a 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. 2b3a2a

2b3a2a1 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). 2b3a2a1

2b3a2a2 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. 2b3a2a2

2b3a2b 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. 2b3a2b

2b3a2b1 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). 2b3a2b1

2b3a2b2 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. 2b3a2b2

2b3a2c 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. 2b3a2c

2b3a2d 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. 2b3a2d

2b3a2e 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. 2b3a2e

2b3a2f 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. 2b3a2f

2b3a2g 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. 2b3a2g

2b3a2h 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. 2b3a2h

2b3b Documentation
--------------- 2b3b

2b3b1 CURRENT ENVIRONMENT 2b3b1

2b3b1a 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. 2b3b1a

2b3b1b 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.) 2b3b1b

2b3b1c 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. 2b3b1c

2b3b1d 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. 2b3b1d

2b3b2 NSW ENVIRONMENT 2b3b2

2b3b2a 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. 2b3b2a

2b3b2a1 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.) 2b3b2a1

2b3b2b 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. 2b3b2b

2b3b2c 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. 2b3b2c

2b3b2c1 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. 2b3b2c1

2b3b2d 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. 2b3b2d

2b3b2e 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. 2b3b2e

2b3b2f 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. 2b3b2f

2b3b2f1 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. 2b3b2f1

2b3b2g 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. 2b3b2g

2b3b2h 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. 2b3b2h

2b3c Office Automation
------------------- 2b3c

2b3c1 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. 2b3c1

2b3c2 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. 2b3c2

2b3c2a 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. 2b3c2a

2b3c2a1 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. 2b3c2a1

2b3c2a2 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. 2b3c2a2

2b3c2b 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. 2b3c2b

2b3c2c 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. 2b3c2c

2b3d General ARPANET Usage
---------------------- 2b3d

2b3d1 CURRENT ENVIRONMENT

Current access to ARPANETed facilities is via the TIP 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. 2b3d1

2b3d2 NSW ENVIRONMENT 2b3d2

2b3d2a 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. 2b3d2a

2b3d2b 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". 2b3d2b

2b3d2c 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. 2b3d2c

2c Rome Air Development Center 2c

2c1 Introduction 2c1

2c1a 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 (Appendix A). 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. 2c1a

2c1b 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. 2c1b

2c1c Five general areas of research have emerged which indicate the thrust of the RADC software research and development effort. 2c1c

2c2 1. MULTICS-NSW Development 2c2

2c2a 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. 2c2a

2c2b 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. 2c2b

2c2c 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. 2c2c

2c2d 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. 2c2d

2c2e The RADC H6180 will be connected to the ARPANET through a hardware interface under procurement from MIT. The interface will connect a host port on the RADC TIP to the multiplexor I/O controller on the H6180. Honeywell will supply the Network Control Program (NCP) and MIT will supply the remaining network software packages . The hardware will be installed in February, and the system should be network operational by March 1975. 2c2e

2c3 2. Structured Programming Environment 2c3

2c3a 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. 2c3a

2c3b 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. 2c3b

2c3c 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. 2c3c

2c3d 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. 2c3d

2c3e 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. 2c3e

2c3f 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. 2c3f

2c3g 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. 2c3g

2c3h 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. 2c3h

2c3i 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. 2c3i

2c3j 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. 2c3j

2c4 3. Language Control Center 2c4

2c4a 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. 2c4a

2c4b 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. 2c4b

2c4c 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. 2c4c

2c4d 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. 2c4d

2c4e 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. 2c4e

2c4f 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. 2c4f

2c4g Compiler Generators 2c4g

2c4g1 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. 2c4g1

2c4h Compiler Validators 2c4h

2c4h1 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. 2c4h1

2c4i Language Statistics Collectors 2c4i

2c4i1 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. 2c4i1

2c4j Language Specification 2c4j

2c4j1 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. 2c4j1

2c4k 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. 2c4k

2c5 4. Document Production Research 2c5

2c5a 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. 2c5a

2c5b 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. 2c5b

2c6 5. WWMCCS Research 2c6

2c6a 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. 2c6a

2c6b 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. 2c6b

2c6b1 Real Time Operating System Techniques 2c6b1

2c6b1a 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. 2c6b1a

2c6b2 Software Reliability and Recovery 2c6b2

2c6b2a 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. 2c6b2a

2c6b3 WWMCCS GCOS Design Improvements 2c6b3

2c6b3a 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. 2c6b3a

2c6b4 Front-end Processing Software 2c6b4

2c6b4a 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. 2c6b4a

2c6b5 WWDMS Investigations 2c6b5

2c6b5a 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. 2c6b5a

2c6c 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. 2c6c

NSW Plan, parts I,II -see comment

(J32766) 17-JUN-75 20:40;;;; Title: Author(s): Lawrence A. Crain/LAC; Distribution: /MAW([ACTION]) NSW([INFO-ONLY]) ; Sub-Collections: NIC NSW; Clerk: LAC; Origin: < DSDC-SYD, NEW-NSW-PLAN.NLS;10, >, 17-JUN-75 20:26 LAC ;;;;####;

1 32766 Distribution

1a Mike A. Wingfield, Jan A. Cornish, Larry L. Garlick, Elizabeth J. Feinler, Kirk Sattley, Ronald P. Uhlig, James B. Lloyd, Frank J. Natoli, Peter C. Waal, Elizabeth K. Michael, William E. Carlson, Steve D. Crocker, David L. Carlstrom, Robert M. Balzer, Richard W. Watson, Lawrence A. Crain, Anthony A.L. Baggiano, Mike A. Wingfield, Jonathan B. Postel, Robert E. Millstein, Duane L. Stone, James E. (Jim) White, Albert J. Mayhan, Albert Vezza, Charles H. Irby, Eugene W. Stubbs, David L. Retz, Stephen T. Walker,

Plan, parts i and II- see comment

LAC 17-JUN-75 21:16 32767

Apparently, in the crash at office-1 last week,wierd things happened to part of the plan. I think I have now recovered the first two parts successfully, but it may be a couple of days befori I get everything back. Please check this over carefully to be sure these parts arn't snashed too. /Larry

Plan, parts i and II- see comment

1 PART I: OVERVIEW OF THE NATIONAL SOFTWARE WORKS PROJECT 1

1a Introduction 1a

1a1 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. 1a1

1a2 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. 1a2

1a3 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. 1a3

1b Background and Technical Need 1b

1b1 The Software Problem (Excerpted from Ref. 2) 1b1

1b1a 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. 1b1a

1b1b 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. 1b1b

1b1c 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. 1b1c

1b1d 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. 1b1d

1b1e 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. 1b1e

1b1f 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. 1b1f

1b1g 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. 1b1g

1b2 Why So Few Tools? 1b2

1b2a 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. 1b2a

1b2b 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. 1b2b

1b2c 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. 1b2c

1b2d 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. 1b2d

1b2e 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. 1b2e

1b3 Origins of the National Software Works Program 1b3

1b3a 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. 1b3a

1b3b 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. 1b3b

1b3c 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. 1b3c

1b3d 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. 1b3d

1b3e 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.) 1b3e

1b3f 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). 1b3f

1b3g 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. 1b3g

1b3h 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. 1b3h

1c National Software Works Design Concepts 1c

1c1 Overview 1c1

1c1a 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. 1c1a

1c1b 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. 1c1b

1c1c 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 10S0 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. 1c1c

1c1d 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. 1c1d

1c1e 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. 1c1e

1c1f 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. 1c1f

1c1g 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. 1c1g

1c2 A New Capability for Project Control 1c2

1c2a 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. 1c2a

1c2b 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. 1c2b

1c2c 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. 1c2c

1c2d 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). 1c2d

1c2e 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). 1c2e

1c2f 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. 1c2f

1c2g 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. 1c2g

1c2h 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. 1c2h

1c2i 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: 1c2i

1c2i1 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. 1c2i1

1c2i2 Project Accountant: This produces reports on the frequency and cost of various patterns of activity interesting to project management. 1c2i2

1c2i3 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. 1c2i3

1c2j 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. 1c2j

1c3 Internal Design 1c3

1c3a 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). 1c3a

1c3a1 (Referencexxx) 1c3a1

1c3b Works Manager 1c3b

1c3b1 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. 1c3b1

1c3b2 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. 1c3b2

1c3b3 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). 1c3b3

1c3c Foreman 1c3c

1c3c1 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. 1c3c1

1c3c2 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. 1c3c2

1c3d Front End 1c3d

1c3d1 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. 1c3d1

1c3d2 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. 1c3d2

1c3d3 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. 1c3d3

1c3d4 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. 1c3d4

1c3d5 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. 1c3d5

1c3d6 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. 1c3d6

1c4 Generalized Support tools 1c4

1c4a Yellow pages 1c4a

1c4a1 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. 1c4a1

1c4b Software libraries. 1c4b

1c4b1 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. 1c4b1

1c4b2 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). 1c4b2

1c4b3 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: 1c4b3

1c4b3a 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. 1c4b3a

1c4b3b The subroutine can be called across the network for debugging, and the user would again be charged a stiff exit fee. 1c4b3b

1c4b3c Only certified linkage tools are allowed to access the subroutine library, and the user is charged each time he builds a new object module. 1c4b3c

1c4b4 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. 1c4b4

1c4b5 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. 1c4b5

1c4c Batch scheduler. 1c4c

1c4c1 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. 1c4c1

1c4c2 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. 1c4c2

1c4c3 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). 1c4c3

1c4c4 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. 1c4c4

1c4d Software conversion aids 1c4d

1c4d1 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. 1c4d1

1c4d2 The NSW environment will help this problem in three ways: 1c4d2

1c4d2a 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. 1c4d2a

1c4d2b The NSW will include a collection of tools for converting production software to run on different (and normally newer) hardware. 1c4d2b

1c4d2c 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. 1c4d2c

1c5 Management Considerations 1c5

1c5a Usage Audit Trail 1c5a

1c5a1 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. 1c5a1

1c5b Maintenance Categories 1c5b

1c5b1 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. 1c5b1

1c5c Cost Predictions and Recovery 1c5c

1c5c1 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. 1c5c1

1c5c2 Efficient use of NSW facilities must be encouraged by charging and reimbursement policies which reflect the true economic value of the services used. 1c5c2

1d Implementaton Guidelines 1d

1d1 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. 1d1

1d2 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. 1d2

1e Participating Organizations 1e

1e1 The NSW Steering Committee is currently composed of representatives of the following government agencies which are supporting the project: 1e1

1e1a Air Force Data Automation Agency(AFDAA) 1e1a

1e1a1 Air Force Data Services Center 1e1a1

1e1a2 Air Force Data Systems Design Center 1e1a2

1e1b Rome Air Development Center 1e1b

1e1c Defense Advanced Research Projects Agency(ARPA) 1e1c

1e2 The following Commercial organizations and universities are under contract to assist in the development of various portions of the system: 1e2

1e2a Applied Data Research(ADR) 1e2a

1e2b Massachusetts Computer Associates (COMPASS) 1e2b

1e2c Stanford Research Institute(SRI) 1e2c

1e2d Bolt Beranek and Newman(BBN) 1e2d

1e2e MIT Project MAC 1e2e

1e2f UCLA Campus Computing Network(CCN) 1e2f

1e2g Speech Communications Research Laboratory(SCRL) 1e2g

1e2h Computer Corporation of America(CCA) 1e2h

1e2i Science Applications, Inc.(SAI) 1e2i

1e3 The NSW Advisory Committee is composed of representatives of the following Government agencies interested in the project: 1e3

1e3a Army Material Command 1e3a

1e3b Headquarters, A.F. Systems Command 1e3b

1e3c *****need a list***** 1e3c

2 PART II: PERSPECTIVE OF PARTICIPATING DOD ORGANIZATIONS' VIEW OF THE NSW 2

2a Air Force Data Services Center 2a

2a1 The AFDSC Environment 2a1

2a1a 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. 2a1a

2a1b 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. 2a1b

2a1c 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. 2a1c

2a1d 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. 2a1d

2a1d1 TABLE 1: AFDSC Resources Devoted to Software Development 2a1d1

2a1d1a FY73 FY74
FY75 2a1d1a

2a1e Non-Multics Development vs Total Manhrs 19.9%
19.5% 15.5% 2a1e

2a1f Multics Dev Total Manhrs 0
3.6% 4.2% 2a1f

2a1g All Dev vs Total Manhrs 19.9%
23.1% 19.7% 2a1g

2a1h 2a1h

2a1i Computer Time For Dev vs All Computer Time 7.5%
7.7% 4.8%
(360/75 and GCOS only) 2a1i

2aij 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. 2aij

2aik 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. 2aik

2a2 AFDSC Operated Computer Service Bureaus 2a2

2a2a 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. 2a2a

2a3 AFDSC Requirements For NSW 2a3

2a3a Usage Scenarios 2a3a

2a3a1 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. 2a3a1

2a3b Tool Requirements, by Target Machine 2a3b

2a3b1 Honeywell G635 or H6000 with GCOS 2a3b1

2a3b1a The primary requirement is for testing batch jobs . Source level interactive debugging is required for COBOL and FORTRAN. 2a3b1a

2a3b1b 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. 2a3b1b

2a3b1c Tools are required for designing, implementing, and restructuring random, ISP, and IDS data bases. 2a3b1c

2a3b1d 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). 2a3b1d

2a3b1e 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. 2a3b1e

2a3b2 IBM 360/370 With OS 2a3b2

2a3b2a There are requirements for FORTRAN, COBOL, and possibly PL/1 development on the 360/370. For each language, source level interactive debugging is required. 2a3b2a

2a3b2b 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. 2a3b2b

2a3b2c 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. 2a3b2c

2a3b2d 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. 2a3b2d

2a3b2e 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. 2a3b2e

2a3b2f 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. 2a3b2f

2a3b3 Multics 2a3b3

2a3b3a 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. 2a3b3a

2a3c Requirements For Tools For Controlling And Maintaining Existing Software Systems. 2a3c

2a3c1 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. 2a3c1

2b Air Force Data Systems Design Center 2b

2b1 THE AFDSDC ENVIRONMENT 2b1

2b1a AFDSDC Mission 2b1a

2b1a1 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. 2b1a1

2b1a1a The only exception to this mission assignment is the personnel function which is the responsibility of the Military Personnel Center. 2b1a1a

2b1a1b 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. 2b1a1b

2b1a2 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. 2b1a2

2b1a3 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

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. 2b3a2f

2b3a2g 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. 2b3a2g

2b3a2h 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. 2b3a2h

2b3b Documentation
--------------- 2b3b

2b3b1 CURRENT ENVIRONMENT 2b3b1

2b3b1a 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 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. 2b1a3

2b1b Organization 2b1b

2b1b1 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. 2b1b1

2b1b1a 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. 2b1b1a

2b1b1b 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. 2b1b1b

2b1b2 A relatively large Auditor staff is co-located with the Center to assure adequacy of audit trails and management during system development. 2b1b2

2b1b3 A Communications detachment from the Air Force Communications Service is also co-located with the Center to provide ongoing communications expertise. 2b1b3

2b1b4 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. 2b1b4

2b1c Responsibilities, Resources, and Workload 2b1c

2b1c1 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 . 2b1c1

2b1c2 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. 2b1c2

2b1c3 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. 2b1c3

2b1c4 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. 2b1c4

2b1c5 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. 2b1c5

2b2 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. 2b2

2b2a 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. 2b2a

2b2b 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. 2b2b

2b2c 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). 2b2c

2b2d 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. 2b2d

2b3 SCENARIOS OF OPERATION: CURRENT AND WITH NSW 2b3

2b3a B3500/B4700 Software Production
------------------------------------ 2b3a

2b3a1 CURRENT ENVIRONMENT 2b3a1

2b3a1a 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. 2b3a1a

2b3a1b 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. 2b3a1b

2b3a2 NSW ENVIRONMENT 2b3a2

2b3a2a 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. 2b3a2a

2b3a2a1 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). 2b3a2a1

2b3a2a2 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. 2b3a2a2

2b3a2b 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. 2b3a2b

2b3a2b1 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). 2b3a2b1

2b3a2b2 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. 2b3a2b2

2b3a2c 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. 2b3a2c

2b3a2d 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. 2b3a2d

2b3a2e 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. 2b3a2e

2b3a2f 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 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. 2b3b1a

2b3b1b 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.) 2b3b1b

2b3b1c 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. 2b3b1c

2b3b1d 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. 2b3b1d

2b3b2 NSW ENVIRONMENT 2b3b2

2b3b2a 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. 2b3b2a

2b3b2a1 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.) 2b3b2a1

2b3b2b 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. 2b3b2b

2b3b2c 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. 2b3b2c

2b3b2c1 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. 2b3b2c1

2b3b2d 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. 2b3b2d

2b3b2e 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. 2b3b2e

2b3b2f 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. 2b3b2f

2b3b2f1 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. 2b3b2f1

2b3b2g 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. 2b3b2g

2b3b2h 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. 2b3b2h

2b3c Office Automation
------------------- 2b3c

2b3c1 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. 2b3c1

2b3c2 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. 2b3c2

2b3c2a 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. 2b3c2a

2b3c2a1 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. 2b3c2a1

2b3c2a2 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. 2b3c2a2

2b3c2b 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. 2b3c2b

2b3c2c 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. 2b3c2c

2b3d General ARPANET Usage
---------------------- 2b3d

2b3d1 CURRENT ENVIRONMENT

Current access to ARPANETed facilities is via the TIP 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. 2b3d1

2b3d2 NSW ENVIRONMENT 2b3d2

2b3d2a 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. 2b3d2a

2b3d2b 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". 2b3d2b

2b3d2c 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. 2b3d2c

2c Rome Air Development Center 2c

2c1 Introduction 2c1

2c1a 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 (Appendix A). 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. 2c1a

2c1b 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. 2c1b

2c1c Five general areas of research have emerged which indicate the thrust of the RADC software research and development effort. 2c1c

2c2 1. MULTICS-NSW Development 2c2

2c2a 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. 2c2a

2c2b 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. 2c2b

2c2c 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. 2c2c

2c2d 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. 2c2d

2c2e The RADC H6180 will be connected to the ARPANET through a hardware interface under procurement from MIT. The interface will connect a host port on the RADC TIP to the multiplexor I/O controller on the H6180. Honeywell will supply the Network Control Program (NCP) and MIT will supply the remaining network software packages . The hardware will be installed in February, and the system should be network operational by March 1975. 2c2e

2c3 2. Structured Programming Environment 2c3

2c3a 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. 2c3a

2c3b 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. 2c3b

2c3c 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. 2c3c

2c3d 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. 2c3d

2c3e 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. 2c3e

2c3f 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. 2c3f

2c3g 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. 2c3g

2c3h 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. 2c3h

2c3i 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. 2c3i

2c3j 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. 2c3j

2c4 3. Language Control Center 2c4

2c4a 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. 2c4a

2c4b 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. 2c4b

2c4c 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. 2c4c

2c4d 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. 2c4d

2c4e 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. 2c4e

2c4f 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. 2c4f

2c4g Compiler Generators 2c4g

2c4g1 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. 2c4g1

2c4h Compiler Validators 2c4h

2c4h1 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. 2c4h1

2c4i Language Statistics Collectors 2c4i

2c4i1 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. 2c4i1

2c4j Language Specification 2c4j

2c4j1 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. 2c4j1

2c4k 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. 2c4k

2c5 4. Document Production Research 2c5

2c5a 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. 2c5a

2c5b 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. 2c5b

2c6 5. WWMCCS Research 2c6

2c6a 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. 2c6a

2c6b 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. 2c6b

2c6b1 Real Time Operating System Techniques 2c6b1

2c6b1a 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. 2c6b1a

2c6b2 Software Reliability and Recovery 2c6b2

2c6b2a 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. 2c6b2a

2c6b3 WWMCCS GCOS Design Improvements 2c6b3

2c6b3a 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. 2c6b3a

2c6b4 Front-end Processing Software 2c6b4

2c6b4a 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. 2c6b4a

2c6b5 WWDMS Investigations 2c6b5

2c6b5a 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. 2c6b5a

2c6c 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. 2c6c

Plan, parts i and II- see comment

(J32767) 17-JUN-75 21:16;;;; Title: Author(s): Lawrence A. Crain/LAC; Distribution: /MAW([ACTION]) NSW([INFO-ONLY]) ; Sub-Collections: NIC NSW; Clerk: LAC; Origin: < DSDC-SYD, NEW-NSW-PLAN.NLS;10, >, 17-JUN-75 20:26 LAC ;;;;####;

1 32767 Distribution

1a Mike A. Wingfield, Jan A. Cornish, Larry L. Garlick, Elizabeth J. Feinler, Kirk Sattley, Ronald P. Uhlig, James B. Lloyd, Frank J. Natoli, Peter C. Waal, Elizabeth K. Michael, William E. Carlson, Steve D. Crocker, David L. Carlstrom, Robert M. Balzer, Richard W. Watson, Lawrence A. Crain, Anthony A.L. Baggiano, Mike A. Wingfield, Jonathan B. Postel, Robert E. Millstein, Duane L. Stone, James E. (Jim) White, Albert J. Mayhan, Albert Vezza, Charles H. Irby, Eugene W. Stubbs, David L. Retz, Stephen T. Walker,

The Whole Universe Catalog: a new tool

1 What it is 1

1a The Whole Universe Catalog (WUC [rhymes with luke]) is a way of looking at things. It consists of a simple accessing system with which you can view from a display or typewriter terminal your personal NLS files as well as the Whole Universe Catalog index. The index currently interfaces to many things of interest to NLS users. If the thing you want is online and you know it's name, WUC can take you to it. If you don't know it's name, you should be able to find it through a synonym or by pointing to categories if it is cataloged in WUC. WUC is meant to eventually become an online university with access to the universe. Hence the ambitious name. 1a

1b The accessing system requires no knowledge of viewspecs, addresses, commandwords, directories or file boundaries. Yet for those who know them, WUC contains all of the addressing and viewing capablities of NLS. It can reduce from ten to one the number of buttons to be pushed for some of the most common viewing functions. From a display terminal it can be used without ever touching the keyboard or keyset. However, any legal NLS ADDRESS can be typed. In DNLS, prompting represents buttons and combinations of buttons on the mouse. Type questionmark in WUC for a short description of the major alternatives. 1b

1c Since WUC has not been accepted as an official ARC program, you cannot type <CTRL-Q> and get help with it or learn about it in help. Instead, type HELP in the wuc command. Also, please send any comments to ident KIRK or sndmsg to KELLEY. Do not send any feedback concerning WUC to FEEDBACK. 1c

2 How to get it 2

2a Use the "Process Branch" command on the following branch (2B) to make WUC available to you. 2a

2b Execute Programs Load Program XPROGRAMS,WUC 2b

2b1 Execute Use IP$_X$PROGRAMS,$_W$UCE$_W$ 2b1

2c In DNLS, push the two rightmost mouse buttons together to go NEXT from where you point and thereby scroll through the rest of this document. 2c

2d CD (CTRL-X) quits WUC. Type ew to "Execute WUC" once you have quit. 2d

3 Features 3

3a With WUC you use the same command for the same viewing function in DNLS as in TNLS. 3a

3b WUC allows viewspecs when you want to input them (as in Jump to Link) but does not prompt for them every time. Only one confirm is necessary because it also specifies your view. The confirm can be thought of as a very special viewspec which also acts to terminate the command. 3b

3c Any NLS address is acceptable but if you wish to address a name in a certain branch, you need not preceed it with exclamation point. 3c

3d If you type a long address and misspell the last word, WUC will take you as far as it can and indicate the rest of the address was not found thus saving you from re-typeing the beginning of the address. 3d

3e Parenthetical comments are not confused with links. Text in parentheses is ignored unless you use the .l address element. 3e

3f Locating Documents 3f

3f1 WUC makes document locator files and super documents easier to use in several ways. Probably the most important to the novice is that you do the same simple things to 1) get to the locator, 2) search down through it, 3) access a particular file and 4) find the piece of information you want in that file. In addition, WUC eliminates five of the rather criptic buttons necessary most of the time in TNLS (pbSN.l<CA><CA>). With WUC in TNLS, you just type the statement number (SN) and confirm. In DNLS this is a single mouse button push. 3f1

3f2 For instance, to access the userguides locator, simply type LOCATOR followed by your CA. If you type LOCATOR OUTPUT you go directly to the output processor guide skipping any view of the locator all together. If you type LOCATOR OUTPUT PES you go directly to the description for the PES output processor directive. In this case all you have to type is OP PES because the output processor guide happens to be available directly from the WUC index under the name OP. 3f2

3g Advantages of WUC over the Help command. 3g

3g1 For help with NLS use WUC and type NLS. You can also go directly to a particular definition or branch of information by typing NLS followed by the name of the information. For example, NLS LINK. 3g1

3g2 WUC contains help for all user-programs made available in WUC by their authors instead of just the ones approved by ARC management. Type PROGRAMS. 3g2

3g3 WUC allows you to scroll to read everything about a certain topic at once without having to back up and try each menu one at a time. When you do want to back up, it is just one push of a button (BC) and in TNLS, only enough is typed to indicate where you have returned. Where you go when you back up is not confusing. Backing up ALWAYS takes you to the next place back. It remembers your last 60 places on a simple stack. 3g3

3g4 You can see everything or just an outline at your discretion. 3g4

3g5 In DNLS, you use the mouse to point to a menu item or point to a word for its definition instead of having to type it in from the keyboard. 3g5

3g6 WUC accepts any valid NLS link, address, viewspec, or content analyzer pattern or program in addition to the simple words accepted by Help. You never get the cryptic message "Illegal list characters" while using WUC. 3g6

3g7 WUC will take you through as many links and files as necessary to get you to where the Help description writer wishes to take you. Even if you type "science fiction" and the information is located under "novels" which is linked to from "fiction" located under "sciences" linked to from "science", WUC will find it. The Help command will not search through links. Also, if the Help database links to a link, you see nothing when you use the Help command. 3g7

3g8 Help Description Writers: you can stop right where you are and edit your current view. 3g8

3g9 Since WUC can access any NLS file with the essential flexibility needed, the current restrictions on format are unnecessary. The ugly, chopped up help database we now have, could be a smoothly flowing online document which would map almost directly into a discursive prose hard copy document. 3g9

3g10 WUC allows you to have your current view printed on the line printer. 3g10

3h Query 3h

3h1 Although it is currently not available for Novice users, it would be very easy to access, for example, the Arpanet Resource

Notebook using WUC. The current "Bring" and "Show" commands as well as the instructions for use duplicated in every file would be unnecessary. Typing questionmark should be sufficient to teach the user what to do. 3h1

3h2 The user could type, for example, NIC SRI-ARC PERSONNEL and see the information directly without having to wade through the current laborious path. 3h2

3h3 All of the writing advantages listed in the Help discussion above are as important for Query database builders. Standard online documents could be written for the database and turned directly into hard copy with very little effort. 3h3

3i If you try to take an invalid link, WUC will indicate the problem and display the statement containing the invalid link. 3i

3j You can go "up" to a higher level from the origin of a file if a link to the source for that file has been specified in the origin. This helps make many files accessible as a single "virtual" file. Such a file is merely a special branch in a theoretically unlimited file and is treated as any other branch. 3j

3k You generally get faster response with WUC because the computer doesn't have to handle as much user interaction before it calls core routines. 3k

3l In WUC, type WUC for a list of the things available in the WUC index. 3l

3m Bon Voyage, information space person. 3m

The Whole Universe Catalog: a new tool

(J32768) 17-JUN-75 21:48;;;; Title: Author(s): Kirk E. Kelley/KIRK; Distribution: /LHD([INFO-ONLY]) GCE([INFO-ONLY] DvN said your might be receptive to this) DAP([INFO-ONLY]) GAS2([INFO-ONLY]) PWO([INFO-ONLY]) DLS([INFO-ONLY]) NJN([INFO-ONLY]) DHC([INFO-ONLY]) WKE([INFO-ONLY]) LPD([INFO-ONLY]) JTM([INFO-ONLY]) DCW([INFO-ONLY]) WSD([INFO-ONLY]) RWW([INFO-ONLY]) JAC3([INFO-ONLY]) DMB([INFO-ONLY]) LLG([INFO-ONLY]) BEV([INFO-ONLY]) ARC-APP([INFO-ONLY]) ; Sub-Collections: SRI-ARC ARC-APP; Clerk: KIRK;

1 32768 Distribution

1a James H. Bair, Robert N. Lieberman, N. Dean Meyer, Sandy L. Johnson, Martin E. Hardy,

1b Lawrence H. Day, Gwen C. Edwards, David A. Potter, Glenn A. Sherwood, Pat Whiting O'Keefe, Duane L. Stone, Nancy J. Neigus, David H. Crocker, William K. English, L. Peter Deutsch, John T. Melvin, Donald C. (Smokey) Wallace, William S. Duvall, Richard W. Watson, Jan A. Cornish, Delorse M. Brooks, Larry L. Garlick, Beverly Boli, Laura J. Metzger, Priscilla A. Wold, Pamela K. Allen, Joan Hamilton, Rene C. Ochoa, Jeffrey C. Peters, Marcia L. Keeney, Jeanne M. Beck, Geoffrey S. Goodfellow, Rodney A. Bondurant, Douglas C. Engelbart, Jeanne M. Leavitt, Susan Gail Roetter, Raymond R. Panko, Adrian C. McGinnis, James C. Norton, J. D. Hopper, Elizabeth J. Feinler

3.12 TPO No. 12 - COMPUTER SYSTEMS TECHNOLOGY

3.12.1 GENERAL OBJECTIVES:

(U) The objective of this TPO is to develop the various computer related technologies which are required to allow computer systems to be used for the efficient and economical solution of numerous Air Force operational problems.

The efforts directed toward meeting this objective are in two major areas - Computer Architecture and Interactive Processing. Computer Architecture deals with the organization of computer components and sub-systems. Interactive Processing addresses the problems of automatic signal classification, man-machine interface, and the automation of operational functions.

The objectives of the efforts in the Computer Architecture area are to: (1) demonstrate the application of Associative Processor (AP) techniques to the solution of Air Force high data rate real-time data processing problems; (2) develop technology which will allow sharing of DOD computing resources such as computers, computer peripherals, data bases and software facilities; (3) develop technology which will make possible the orderly design, test and evaluation of special computer architecture required by the Air Force before the architecture design is given to a contractor to build.

Efforts in the Interactive Processing area are directed toward: (1) developing the facilities and the expertise required to solve the complex data processing problems of automatic signal classification technique development, and assisting a wide variety of users in actually developing automatic classification and identification equipment for use in various Air Force surveillance systems; (2) developing cost effective interactive graphics techniques and systems which will lower Air Force operational costs and increase effectiveness.

3.12.2 SPECIFIC GOALS AND TECHNICAL APPROACHES:

(U) The plan for attaining the TPO objective is shown in the overview chart (Page 3.12-2). The technical efforts covered by this TPO are oriented toward providing the four major products shown..IFirst=10;

3.12.2.1 (U) ASSOCIATIVE PROCESSOR COMPUTING SYSTEM - The goal of this product is to provide techniques capable of meeting the ever increasing data processing requirements of the Air Force as typified by the high data rate real-time processing demands imposed by systems such as the Airborne Warning and Control System (AWACS).

As indicated by the Milestone Chart (Page 3.12-), this goal is being achieved by RADC's Associative Processor Testbed facility. The testbed facility consists of an Associative Processor tied to RADC's HIS 645 sequential computer, a situation display capability for purposes of providing operator interaction with the executing programs, and equipment for monitoring the performance of the testbed equipment while executing operational type programs. Completion of the testbed system is planned during FY-76, when a data manipulation capability for effecting high utilization of processing elements will be installed to complement the interim capabilities available at the present time.

The major on-going effort for assessing the capability of an associative processor to meet real-time requirements is directed toward the AWACS requirements. Associative Processor Application Software, to accomplish the functions required by AWACS, is being developed for analysis and demonstration on the AP testbed. This activity, which is being accomplished both in-house and contractually, will concentrate on the data processing aspects of active and passive tracking, radar data correlation, display processing, and signal processing. Completion is expected in mid FY-76. Major improvements in capability are envisioned for AWACS-like systems using an AP approach rather than a sequential computer. AP active tracking tests to date indicate a capability of handling five times as many radar tracks. Efforts to develop software for other Air Force system functions requiring real time processing capability will be started upon completion of the AWACS work. In addition, Syracuse University is developing software to perform functions such as matrix manipulations, fast fourier transforms, partial differential equation solutions, and traffic control/collision predictions, and MITRE Corporation is developing software in functional areas directly applicable to ESD project requirements.

provides solution to the problem. The design which is a by-product of the configured system will be given to a contractor to build only after it has been tested and evaluated using significant benchmark software. After the design has been released to the contractor, the "configured architecture" will then serve as an interim computer so that software development can be continued while the special computer is being built.

The configurable processor concept is an outgrowth of the associative processor program, the microprocessor technology developments which now provide complete computer functions on a single integrated circuit chip, and the advances which have been made in the microprogramming technology.

The configurable processor program will begin with the purchase of off-the-shelf Micro Function Components including microfunction chips, a microprogrammable control unit and a multi-dimensional access memory. These components will be assembled into an extremely inexpensive system which through firmware can be made to act as any architectural form (sequential processor, parallel processor, multi-processor, etc.) having any desired word/logic length. The Configurable Processor Design, Test and Evaluation will be accomplished on these assembled components and as such will represent a Configurable Processor Testbed Facility.

Configurable Processor Application Software will be developed to demonstrate the process of structuring the configurable processor based on: (1) the structure of the problem, and (2) the structure of the software for implementing the problem solution.

The end product will be: (1) an inexpensive, Air Force Configurable Processor Testbed Facility, (2) utility software for facilitating the configuration process, and (3) validation data which will demonstrate the process and the benefits that a Configurable Processor Facility brings to the Air Force.

3.12.2.3 (U) SIGNAL CLASSIFICATION - The automatic classification of signals is required to support many Air Force objectives including space defense, remote surveillance, base defense, and tactical interdiction. Devices (signal classifiers) which analyze the output electronic signals from the transducers of various sensor systems, including radar, photometric, seismic, acoustic, etc., and automatically recognize these signals as occurring in a specific target class, must be developed. These classifiers will increase the effectiveness of the sensor system since false alarms will be eliminated/reduced and will allow more accurate and rapid assessment of the threat by commanders. Classifiers may be implemented as part of a data processing system, be part of other ground or airborne installations, or be contained in sensors impact implanted into enemy territory.

The technical approach to the design of these classifiers is to use advanced pattern recognition techniques (software) residing on a general purpose computer interactively coupled to a high performance graphics console to provide general purpose design tools. These tools allow the design engineer to analyze a data set, extract features characteristic of each target class, evaluate these features and design the target classifier. Since several solutions can be developed rapidly, the most cost effective in terms of accuracy versus complexity (i.e., the number and type of features or method of classification required) can be implemented.

Up to this point, development work has centered around the tools and techniques to allow problem solution. With the validity of our interactive pattern recognition approach having been verified on specific previous applications, our attention for FY-76 and beyond is on the implementation of a fully operational Pattern Recognition Design Facility, a unique capability within DOD and industry. This facility is needed in order to support the future signal classification workload as a production oriented capability, addressed to specific user requirements. In order to arrive at this posture in a timely fashion, the Milestone Chart (Page 3.12-) line items associated with waveform processing and preprocessing and with the updating of the On-Line Pattern Analysis and Recognition System (OLPARS) must be completed.

Waveform Preprocessing Techniques encompass those ancillary projects necessary for a viable system. In FY-74, such items as hybrid feature extraction software, hybrid preprocessing routines, and the configuring of a peripheral image dissector, permitted the handling of digital, analog, and photographic waveform data input. In FY-75, the item will be expanded to include a data

During FY-75, the Associative Processor testbed will become a remotely accessible Associative Processing Research Facility. Under this mode of operation, DOD agencies and contractors will be able to enter batch jobs from remote stations on a twenty hour per day basis.

Two AP design studies are being conducted in support of RADC TPO-9 "Telecommunications". This TPO supports improvements to the Defense Communications System (DCS). The first is the Communications Processing System study which will examine various architectures, including the Associative Processor, in the design of a processor to handle all types of communications traffic both digital and analog. Completion is planned for mid FY-75. The second design study is for an associative communications multiplexer which will compress and multiplex digital signals using an Associative Processor. Completion is planned for the end of FY-75.

The prime end goal of the Associative Processor Project is the development of hardware and software specifications for a cost-effective, highly reliable Associative Processor Computing System (APCS) suitable for airborne and tactical environments. These specifications and the rationale developed by the project during the 1973-1976 time period are expected to lead to a 6.4 program to actually fabricate the evolved APCS design. This program, estimated at 14 million dollars, will result in a militarized Associative Processor and its supporting software directed toward applications such as AWACS and the Advanced Airborne Command Post. The computer system will serve as the basis for continued software development and application programming at RADC and can be used as a prototype design for operational Air Force systems. Target date for completion is FY-79.

3.12.2.2 (U) DISTRIBUTED COMPUTATION SYSTEMS - The objective of this Product is to provide the Air Force with mechanisms for maximizing the utilization of its computing resources. Distributed Computation Systems are of two types: (1) computer systems which are physically distributed but tied together by means of a network such as the ARPANET, and (2) computer systems which have, on a single mainframe, a variety of functionally distributed capabilities which can strategically be brought together in an optimum fashion to provide highly efficient/low cost solutions to the

Air Force real time/special requirements. Milestone Chart (Page 3.12-) covers both types of Distributed Computation Systems.

To achieve the first objective of providing the capability to efficiently and effectively utilize the dispersed computational resources inherently present in a distributed heterogeneous computer network, the Distributed Computation Techniques and Applications effort will identify problems inherent in present netted computer systems which deter resource sharing among computers. It will explore existing software and hardware systems to determine what fixes can be made to overcome the problem with these systems. It will then propose possible system architectures which should lead to a system which allows efficient resource sharing in a manner which makes the mechanics of network use completely transparent to the network resource user. It will implement the software fixes proposed by the study and test them in actual operational systems. Any additional techniques which appear promising will also be explored.

The System Architecture effort will involve simulation and, if possible, implementation of the architecture proposed by the Distributed Computation Techniques and Applications study. These will be thoroughly tested to determine their applicability to actual operational systems such as the World Wide Military Command and Control System (WWMCCS).

The AFSC Network effort will develop the interfaces to connect 16 AFSC computers to the ARPANET in two phases. In the first phase, three CDC-6600 machines at ASD, ADTC and AFWL will be connected to the net. In the second phase, the remaining 13 sites will be connected.

The second objective of the Distributed Computation Systems product area is to develop an effective low cost approach for producing cost effective computing capability required to solve special processing functions which are peculiar to the Air Force and which cannot be satisfied by computers existing in government and industries inventories. For this purpose, a program called Configurable Processor has been initiated. A Configurable Processor is a functionally distributed computation system which can be structured by means of firmware to assume the optimal, cost effective architecture form required by the problem and the software which link from the RADC HIS 6180 to the PDP 11/45 computer, so that each computer may be used to the greatest advantage.

The Waveform Processing System (WPS) addresses the problem of quickly and efficiently extracting discriminatory feature information from the waveform data. When completed in early FY-77, it will allow the analyst to dynamically interact with the data in defining and testing candidate feature sets. The result will be a more effective design in a minimum of time.

Improvements in the prototype version of OLPARS are of two categories: (1) the implementation of the logic design capability on the PDP 11/45 computer resident with the WPS, and (2) the implementation of the logic design under the MULTICS operating system on the RADC HIS 6180 computer. In the first OLPARS version, full advantage will be taken of the high performance graphics terminals associated with the PDP 11/45 system, and of the interplay between the feature extraction and logic design software. The MULTICS/OLPARS version will take advantage of the inherent flexibility of the MULTICS operating system and provide, for the first time, a multi-user capability for pattern recognition applications. If RADC becomes a host on the ARPA network, the multiple users could be from any agency with access to the network. By this method, we not only produce results in the terms of specific classifier designs for certain users, we also will be able to export our software capability to those users who would like to do their own analysis.

Using the capability of the interim Pattern Recognition Design Facility which currently exists, RADC is pursuing a vigorous program in selected applications areas to support user signal classification requirements. The Aerospace Shape Identification (Radar) line item includes a study of the feasibility of recognizing characteristic shapes of aircraft and space objects by analyzing the narrow pulse radar returns. With sufficiently descriptive features which are independent of orientation, one would dramatically decrease the storage and speed requirements for any classification system.

In a related area of Photometric Signature Analysis, RADC is studying actual space object signatures derived from telescopically collected reflected sunlight in order to determine certain characteristic information. In the first part of this effort, in direct support of SAMSO's Satellite Observables Program (per TN-SAMSO-RADC-1708-73-01), the analysis is performed on selected satellite signatures as determined by SAMSO. In the second portion, the photometric data collected on various selected space objects will be analyzed to determine features and classification logic for automatic discrimination purposes. This second portion is in support of the Air Force Avionics Laboratory and is directed initially toward tumbling rocket bodies. Results of both the photometric and aerospace shape work are directly applicable to the needs of ADC as expressed in ADC-ROC 16-71.

Another major applications area, represented by the Acoustic/Seismic/Magnetic Data Analysis area, includes projects which deal with the processing of ground based analog sensor data. In each case, the project involves aspects of signal processing, to determine sensor and event characteristics, and the application of classical pattern recognition to design discrimination logic.

The first of these projects involves the processing of aircraft acoustic signatures for the purposes of remotely monitoring air base activity. This project, in support of RADC TPO-7 "Tactical Surveillance and Control", involves the collection and analysis of a large aircraft signature data base of take-offs, landings, and fly-overs. Logic will be designed to discriminate between the aircraft events of interest and thereby monitor the base activity. The discrimination logic in this application, as in all of the others, initially is in the form of software programs and equations, but can be implemented in microcircuits and integrated with the sensor electronics as required.

Another project, in support of ESD Advanced Development Program 681E and RADC TPO-7 "Tactical Surveillance and Control", involves the modernization of line sensors to be used in the Base Installation Security System. The sensor outputs consist of magnetic and seismic waveforms which are generated by valid intrusions and by false alarm events. The modernization task involves the analysis of the sensor waveforms and the design of discrimination logic to sort out valid intrusions from the false alarms, such as wind and lightning. Once a desirably low false alarm rate has been achieved, these improved sensors based on RADC's new design classification logic will be acquired by ESD for installation at air bases throughout the world.

The last of the projects being pursued in the ground-based sensor area is the Nuclear Data Analysis project in support of the Arms Control and Disarmament Agency (ACDA). In this project, short period seismograms of teleseismic Eurasian events will be analyzed with the objective of automatically classifying earthquakes from underground nuclear explosions. Using the extensive event data base collected by ACDA, RADC will analyze the signals, extract the discriminatory information and design classification logic for the task.

The last of the applications areas to be discussed is one represented by the Milestone Chart line item entitled Speaker Identification. In support of RADC TPO-2 "Intelligence Analysis and Exploitation", a data set of human speech is being analyzed. The analysis is focused on determining, and if possible, quantifying inter-speaker similarities and intra-speaker differences for application in security related speaker recognition devices.

New pattern recognition techniques for feature extraction and target classification are continually being evaluated and the potentially useful ones added to those tools so that they evolve to solve an ever increasing spectrum of problems. Development in this area is expected to continue through at least FY-77.

3.12.2.4 (U) COMPUTER GRAPHICS SYSTEMS - The goal of the computer graphics technology area is to provide cost effective solutions to the problems encountered in developing low cost multi-terminal computer graphics systems which will present computer generated data to an operator and allow him to interact with it. The data presented can be in the form of alphanumeric symbology or line drawn graphics. The system must allow the operator to interact with the data base and modify or control the overall processing function by means of his input devices while observing the results with some type of display. This technology and the technology being developed under RADC TPO-4 "Intelligence Data Handling", fully complement each other since TPO-4 efforts are directed toward image analysis which is not a goal of this TPO.

One of the most effective types of display presentation is the interactive graphics display. In this mode, the operator has access to and control over the computing process with the results presented to him in the form of graphical figures. There are numerous levels of performance and cost associated with each system, as well as a variety of possible implementation schemes. The selected system must represent a balance between performance, cost and flexibility.

The RADC computer graphics program is a combination of both contractual and in-house efforts. All of these efforts depend heavily on the interactive processing facility recently installed at RADC. This facility includes a flexible computing system as well as high performance graphics and allows the operator to both perform development functions as well as to simulate several applications functions for specific operational users. The Milestone Chart (Page 3.12-) outlines the computer graphics program.

The Air Force Global Weather Central (GWC) Interactive Graphics System effort is directed toward providing their analysts and forecasters with the ability to generate, edit, analyze and update various weather charts and forecasts in the interactive mode. This will increase the flexibility as well as decrease the cost of the various operations involved. The final system configuration will include numerous terminals and several forecast locations as well as a direct hook-up to the main GWC computer facility. This is a multi-phase, multi-year effort which will result in the automation of a significant portion of the GWC operation. This will be accomplished through the implementation of several mini-computer based graphics systems, as well as the development of specialized software and techniques to handle the particular characteristics of the environmental data. The effort is being pursued in support of MAC-DAR-L72-7 with full implementation scheduled for FY-78.

One of the prime considerations in applying computer graphics is that of economy. The specific approach that will reduce the per-terminal cost of multiple terminal graphics systems is the use of raster (or digital TV) techniques. Such a system not only has the potential for high performance at relatively low cost, but also offers the unique feature of being able to mix computer generated graphical data with pictorial information. The Raster Graphic Techniques effort is aimed at investigating the feasibility of performing the graphics functions using interactive TV displays. The goal is to provide performance equivalent to present selective address systems with a significant reduction in cost. The culmination of the program in FY-78 will be a Multi-Terminal Raster

Graphics System which demonstrates equivalent performance at much lower cost than previously available systems.

The Graphics Software Techniques area represents a group of related and continuing efforts aimed at decreasing the complexity and increasing the efficiency of graphics software and the means to generate it. To provide a testbed within which to work, a complete graphics software system has been configured to operate on our graphics facility. Among the areas which will be pursued in future development efforts are the investigation of graphic data base structures and their effect on processing efficiency, as well as the practicality and desirability of including a graphic extension within a higher order language, particularly the block structured declarative type.

The Graphics Modeling effort is aimed at the development of a modeling system to investigate the various parameters utilized in the design of interactive graphic systems. These include the use of satellite processors, distributed as opposed to concentrated data bases, the effect of response time on various interactive devices, and operator performance within such a system. The result will be more effective and efficient system designs while minimizing the overall cost.

3.12.3 RELATED EFFORTS:

(U) Numerous efforts which are related to this TPO are being pursued by other Air Force Laboratories and other agencies, both within DOD and in industry. Close technical liaison is being maintained on these efforts and any technical advances made will be applied to this TPO. Conversely, the results of efforts under this TPO are made available to other agencies for application to their programs. Following is a summary of the major efforts related to the products of this TPO.

ASSOCIATIVE PROCESSOR COMPUTING SYSTEM - The Federal Aviation Agency is studying the use of an associative processor for air traffic control processing. A prototype STARAN was used for conflict prediction studies at the Knoxville, Tennessee test site. Further studies are being conducted with three contractors, each applying benchmark problems to their machine. The Defense Mapping Agency

(DMA) is leasing an associative processor to study its applicability to the post processing functions of cartography.

DISTRIBUTED COMPUTATIONAL SYSTEM - The ARPANET designed and implemented by the Defense Advanced Research Projects Agency (ARPA) embodies many of the prerequisite features which will provide the basis for this advanced study. Specifically, it includes all the protocols which allow transfer of data and programs between computers and between peripherals and computers. The National Software Works program, sponsored by ARPA in conjunction with RADC TPO-11 "Software Sciences Technology", having the objective of giving a programmer access to many software development tools, will coordinate closely with and use many of the results of this program.

CONFIGURABLE PROCESSOR FACILITY - Univac has a program under IR and D which will be used to design their next generation computer systems. They intend to build a microprogrammed multi-processor with an extremely flexible internal communications scheme to support the program. IBM, under internal corporate research, is designing a direct execution machine (a machine which directly executes higher order language code).

TARGET RECOGNITION - The Army is applying pattern recognition techniques to speaker identification photograph analysis, shock trauma analysis and mine fusing applications. The Navy is using pattern recognition for speech recognition and for sonar target identification. NASA is doing work in identification of various earth resources using multi-spectral scanners. The Arms Control and Disarmament Agency is utilizing pattern recognition techniques to identify nuclear blast detonations.

It should be noted that the preceding work has been done in close cooperation with RADC. Some of the work was actually completed on the RADC system by other services after training and familiarization by RADC personnel. These agencies have either begun development or are considering development of a research tool patterned after the RADC On-Line Pattern Analysis and Recognition System (OLPARS); or are planning to continue using the RADC system.

COMPUTER GRAPHICS SYSTEMS - There are numerous efforts in computer graphics being pursued both within DOD and in other government agencies. The Army is developing cartographic analysis techniques using interactive graphics while the Navy is synthesizing radar backgrounds and clutter to test radar signal processing systems. NASA has used computer graphics extensively for both computer aided design and flight simulation. More recently within the National Oceanographic and Atmospheric Administration computer graphics has been applied to the analysis of meteorological data particularly relative to severe weather conditions. ARPA has been one of the prime movers in the development of graphic techniques primarily through its university contracts.

3.12.4 REQUIREMENTS:

(U) There are numerous Air Force requirements for the technologies being developed under this TPO. High speed data processing techniques, software first techniques, automatic target identification techniques, and interactive graphics systems are being developed to satisfy the following requirements:

tpo 12 - 1974

(J32769) 18-JUN-75 05:27;;;; Title: Author(s): Roberta J. Carrier/RJC; Sub-Collections: NIC; Clerk: RJC; Origin: <LORETO>TPO12.NLS;1, 21-AUG-74 07:47 DRL2 ;####;

3.12 TPO No. 12 - COMPUTER SYSTEMS TECHNOLOGY

3.12.1 GENERAL OBJECTIVES:

(U) The objective of this TPO is to develop the various computer related technologies which are required to allow computer systems to be used for the efficient and economical solution of numerous Air Force operational problems.

The efforts directed toward meeting this objective are in two major areas - Computer Architecture and Interactive Processing. Computer Architecture deals with the organization of computer components and sub-systems. Interactive Processing addresses the problems of automatic signal classification, man-machine interface, and the automation of operational functions.

The objectives of the efforts in the Computer Architecture area are to: (1) demonstrate the application of Associative Processor (AP) techniques to the solution of Air Force high data rate real-time data processing problems; (2) develop technology which will allow sharing of DOD computing resources such as computers, computer peripherals, data bases and software facilities; (3) develop technology which will make possible the orderly design, test and evaluation of special computer architecture required by the Air Force before the architecture design is given to a contractor to build.

Efforts in the Interactive Processing area are directed toward: (1) developing the facilities and the expertise required to solve the complex data processing problems of automatic signal classification technique development, and assisting a wide variety of users in actually developing automatic classification and identification equipment for use in various Air Force surveillance systems; (2) developing cost effective interactive graphics techniques and systems which will lower Air Force operational costs and increase effectiveness.

3.12.2 SPECIFIC GOALS AND TECHNICAL APPROACHES:

(U) The plan for attaining the TPO objective is shown in the overview chart (Page 3.12-2). The technical efforts covered by this TPO are oriented toward providing the four major products shown.

3.12.2.1 (U) ASSOCIATIVE PROCESSOR COMPUTING SYSTEM - The goal of this product is to provide techniques capable of meeting the ever increasing data processing requirements of the Air Force as typified by the high data rate real-time processing demands imposed by systems such as the Airborne Warning and Control System (AWACS).

As indicated by the Milestone Chart (Page 3.12-), this goal is being achieved by RADC's Associative Processor Testbed facility. The testbed facility consists of an Associative Processor tied to RADC's HIS 645 sequential computer, a situation display capability for purposes of providing operator interaction with the executing programs, and equipment for monitoring the performance of the testbed equipment while executing operational type programs. Completion of the testbed system is planned during FY-76, when a data manipulation capability for effecting high utilization of processing elements will be installed to complement the interim capabilities available at the present time.

The major on-going effort for assessing the capability of an associative processor to meet real-time requirements is directed toward the AWACS requirements. Associative Processor Application Software, to accomplish the functions required by AWACS, is being developed for analysis and demonstration on the AP testbed. This activity, which is being accomplished both in-house and contractually, will concentrate on the data processing aspects of active and passive tracking, radar data correlation, display processing, and signal processing. Completion is expected in mid FY-76. Major improvements in capability are envisioned for AWACS-like systems using an AP approach rather than a sequential computer. AP active tracking tests to date indicate a capability of handling five times as many radar tracks. Efforts to develop software for other Air Force system functions requiring real time processing capability will be started upon completion of the AWACS work. In addition, Syracuse University is developing software to perform functions such as matrix manipulations, fast fourier transforms, partial differential equation solutions, and traffic control/collision predictions, and MITRE Corporation is developing software in functional areas directly applicable to ESD project requirements.

During FY-75, the Associative Processor testbed will become a remotely accessible Associative Processing Research Facility. Under this mode of operation, DOD agencies and contractors will be able to enter batch jobs from remote stations on a twenty hour per day basis.

Two AP design studies are being conducted in support of RADC TPO-9 "Telecommunications". This TPO supports improvements to the Defense Communications System (DCS). The first is the Communications Processing System study which will examine various architectures, including the Associative Processor, in the design of a processor to handle all types of communications traffic both digital and analog. Completion is planned for mid FY-75. The second design study is for an associative communications multiplexer which will compress and multiplex digital signals using an Associative Processor. Completion is planned for the end of FY-75.

The prime end goal of the Associative Processor Project is the development of hardware and software specifications for a cost-effective, highly reliable Associative Processor Computing System (APCS) suitable for airborne and tactical environments. These specifications and the rationale developed by the project during the 1973-1976 time period are expected to lead to a 6.4 program to actually fabricate the evolved APCS design. This program, estimated at 14 million dollars, will result in a militarized Associative Processor and its supporting software directed toward applications such as AWACS and the Advanced Airborne Command Post. The computer system will serve as the basis for continued software development and application programming at RADC and can be used as a prototype design for operational Air Force systems. Target date for completion is FY-79.

3.12.2.2 (U) DISTRIBUTED COMPUTATION SYSTEMS - The objective of this product is to provide the Air Force with mechanisms for maximizing the utilization of its computing resources. Distributed Computation Systems are of two types: (1) computer systems which are physically distributed but tied together by means of a network such as the ARPANET, and (2) computer systems which have, on a single mainframe, a variety of functionally distributed capabilities which can strategically be brought together in an optimum fashion to provide highly efficient/low cost solutions to the

Air Force real time/special requirements. Milestone Chart (Page 3.12-) covers both types of Distributed Computation Systems.

To achieve the first objective of providing the capability to efficiently and effectively utilize the dispersed computational resources inherently present in a distributed heterogeneous computer network, the Distributed Computation Techniques and Applications effort will identify problems inherent in present netted computer systems which deter resource sharing among computers. It will explore existing software and hardware systems to determine what fixes can be made to overcome the problem with these systems. It will then propose possible system architectures which should lead to a system which allows efficient resource sharing in a manner which makes the mechanics of network use completely transparent to the network resource user. It will implement the software fixes proposed by the study and test them in actual operational systems. Any additional techniques which appear promising will also be explored.

The System Architecture effort will involve simulation and, if possible, implementation of the architecture proposed by the Distributed Computation Techniques and Applications study. These will be thoroughly tested to determine their applicability to actual operational systems such as the World Wide Military Command and Control System (WWMCCS).

The AFSC Network effort will develop the interfaces to connect 16 AFSC computers to the ARPANET in two phases. In the first phase, three CDC-6600 machines at ASD, ADTC and AFWL will be connected to the net. In the second phase, the remaining 13 sites will be connected.

The second objective of the Distributed Computation Systems product area is to develop an effective low cost approach for producing cost effective computing capability required to solve special processing functions which are peculiar to the Air Force and which cannot be satisfied by computers existing in government and industries inventories. For this purpose, a program called Configurable Processor has been initiated. A Configurable Processor is a functionally distributed computation system which can be structured by means of firmware to assume the optimal, cost effective architecture form required by the problem and the software which provides solution to the problem. The design which is a by-product of the configured system will be given to a contractor to build only after it has been tested and evaluated using significant benchmark software. After the design has been released to the contractor, the "configured architecture" will then serve as an interim computer so that software development can be continued while the special computer is being built.

The configurable processor concept is an outgrowth of the associative processor program, the microprocessor technology developments which now provide complete computer functions on a single integrated circuit chip, and the advances which have been made in the microprogramming technology.

The configurable processor program will begin with the purchase of off-the-shelf Micro Function Components including microfunction chips, a microprogrammable control unit and a multi-dimensional access memory. These components will be assembled into an extremely inexpensive system which through firmware can be made to act as any architectural form (sequential processor, parallel processor, multi-processor, etc.) having any desired word/logic length. The Configurable Processor Design, Test and Evaluation will be accomplished on these assembled components and as such will represent a Configurable Processor Testbed Facility.

Configurable Processor Application Software will be developed to demonstrate the process of structuring the configurable processor based on: (1) the structure of the problem, and (2) the structure of the software for implementing the problem solution.

The end product will be: (1) an inexpensive, Air Force Configurable Processor Testbed Facility, (2) utility software for facilitating the configuration process, and (3) validation data which will demonstrate the process and the benefits that a Configurable Processor Facility brings to the Air Force.

3.12.2.3 (U) SIGNAL CLASSIFICATION - The automatic classification of signals is required to support many Air Force objectives including space defense, remote surveillance, base defense, and tactical interdiction. Devices (signal classifiers) which analyze the output electronic signals from the transducers of various sensor systems, including radar, photometric, seismic, acoustic, etc., and automatically recognize these signals as occurring in a specific target class, must be developed. These classifiers will increase the effectiveness of the sensor system since false alarms will be eliminated/reduced and will allow more accurate and rapid assessment of the threat by commanders. Classifiers may be implemented as part of a data processing system, be part of other ground or airborne installations, or be contained in sensors impact implanted into enemy territory.

The technical approach to the design of these classifiers is to use advanced pattern recognition techniques (software) residing on a general purpose computer interactively coupled to a high performance graphics console to provide general purpose design tools. These tools allow the design engineer to analyze a data set, extract features characteristic of each target class, evaluate these features and design the target classifier. Since several solutions can be developed rapidly, the most cost effective in terms of accuracy versus complexity (i.e., the number and type of features or method of classification required) can be implemented.

Up to this point, development work has centered around the tools and techniques to allow problem solution. With the validity of our interactive pattern recognition approach having been verified on specific previous applications, our attention for FY-76 and beyond is on the implementation of a fully operational Pattern Recognition Design Facility, a unique capability within DOD and industry. This facility is needed in order to support the future signal classification workload as a production oriented capability, addressed to specific user requirements. In order to arrive at this posture in a timely fashion, the Milestone Chart (Page 3.12-) line items associated with waveform processing and preprocessing and with the updating of the On-Line Pattern Analysis and Recognition System (OLPARS) must be completed.

Waveform Preprocessing Techniques encompass those ancillary projects necessary for a viable system. In FY-74, such items as hybrid feature extraction software, hybrid preprocessing routines, and the configuring of a peripheral image dissector, permitted the handling of digital, analog, and photographic waveform data input. In FY-75, the item will be expanded to include a data link from the RADC HIS 6180 to the PDP 11/45 computer, so that each computer may be used to the greatest advantage.

The Waveform Processing System (WPS) addresses the problem of quickly and efficiently extracting discriminatory feature information from the waveform data. When completed in early FY-77, it will allow the analyst to dynamically interact with the data in defining and testing candidate feature sets. The result will be a more effective design in a minimum of time.

Improvements in the prototype version of OLPARS are of two categories: (1) the implementation of the logic design capability on the PDP 11/45 computer resident with the WPS, and (2) the implementation of the logic design under the MULTICS operating system on the RADC HIS 6180 computer. In the first OLPARS version, full advantage will be taken of the high performance graphics terminals associated with the PDP 11/45 system, and of the interplay between the feature extraction and logic design software. The MULTICS/OLPARS version will take advantage of the inherent flexibility of the MULTICS operating system and provide, for the first time, a multi-user capability for pattern recognition applications. If RADC becomes a host on the ARPA network, the multiple users could be from any agency with access to the network. By this method, we not only produce results in the terms of specific classifier designs for certain users, we also will be able to export our software capability to those users who would like to do their own analysis.

Using the capability of the interim Pattern Recognition Design Facility which currently exists, RADC is pursuing a vigorous program in selected applications areas to support user signal classification requirements. The Aerospace Shape Identification (Radar) line item includes a study of the feasibility of recognizing characteristic shapes of aircraft and space objects by analyzing the narrow pulse radar returns. With sufficiently descriptive features which are independent of orientation, one would dramatically decrease the storage and speed requirements for any classification system.

In a related area of Photometric Signature Analysis, RADC is studying actual space object signatures derived from telescopically collected reflected sunlight in order to determine certain characteristic information. In the first part of this effort, in direct support of SAMSO's Satellite Observables Program (per TN-SAMSO-RADC-1708-73-01), the analysis is performed on selected satellite signatures as determined by SAMSO. In the second portion, the photometric data collected on various selected space objects will be analyzed to determine features and classification logic for automatic discrimination purposes. This second portion is in support of the Air Force Avionics Laboratory and is directed initially toward tumbling rocket bodies. Results of both the photometric and aerospace shape work are directly applicable to the needs of ADC as expressed in ADC-ROC 16-71.

Another major applications area, represented by the Acoustic/Seismic/Magnetic Data Analysis area, includes projects which deal with the processing of ground based analog sensor data. In each case, the project involves aspects of signal processing, to determine sensor and event characteristics, and the application of classical pattern recognition to design discrimination logic.

The first of these projects involves the processing of aircraft acoustic signatures for the purposes of remotely monitoring air base activity. This project, in support of RADC TPO-7 "Tactical Surveillance and Control", involves the collection and analysis of a large aircraft signature data base of take-offs, landings, and fly-overs. Logic will be designed to discriminate between the aircraft events of interest and thereby monitor the base activity. The discrimination logic in this application, as in all of the others, initially is in the form of software programs and equations, but can be implemented in microcircuits and integrated with the sensor electronics as required.

Another project, in support of ESD Advanced Development Program 681E and RADC TPO-7 "Tactical Surveillance and Control", involves the modernization of line sensors to be used in the Base Installation Security System. The sensor outputs consist of magnetic and seismic waveforms which are generated by valid intrusions and by false alarm events. The modernization task involves the analysis of the sensor waveforms and the design of discrimination logic to sort out valid intrusions from the false alarms, such as wind and lightning. Once a desirably low false alarm rate has been achieved, these improved sensors based on RADC's new design classification logic will be acquired by ESD for installation at air bases throughout the world.

The last of the projects being pursued in the ground-based sensor area is the Nuclear Data Analysis project in support of the Arms Control and Disarmament Agency (ACDA). In this project, short period seismograms of teleseismic Eurasian events will be analyzed with the objective of automatically classifying earthquakes from underground nuclear explosions. Using the extensive event data base collected by ACDA, RADC will analyze the signals, extract the discriminatory information and design classification logic for the task.

The last of the applications areas to be discussed is one represented by the Milestone Chart line item entitled Speaker Identification. In support of RADC TPO-2 "Intelligence Analysis and Exploitation", a data set of human speech is being analyzed. The analysis is focused on determining, and if possible, quantifying inter-speaker similarities and intra-speaker differences for application in security related speaker recognition devices.

New pattern recognition techniques for feature extraction and target classification are continually being evaluated and the potentially useful ones added to those tools so that they evolve to solve an ever increasing spectrum of problems. Development in this area is expected to continue through at least FY-77.

3.12.2.4 (U) COMPUTER GRAPHICS SYSTEMS - The goal of the computer graphics technology area is to provide cost effective solutions to the problems encountered in developing low cost multi-terminal computer graphics systems which will present computer generated data to an operator and allow him to interact with it. The data presented can be in the form of alphanumeric symbology or line drawn graphics. The system must allow the operator to interact with the data base and modify or control the overall processing function by means of his input devices while observing the results with some type of display. This technology and the technology being developed under RADC TPO-4 "Intelligence Data Handling", fully complement each other since TPO-4 efforts are directed toward image analysis which is not a goal of this TPO.

One of the most effective types of display presentation is the interactive graphics display. In this mode, the operator has access to and control over the computing process with the results presented to him in the form of graphical figures. There are numerous levels of performance and cost associated with each system, as well as a variety of possible implementation schemes. The selected system must represent a balance between performance, cost and flexibility.

The RADC computer graphics program is a combination of both contractual and in-house efforts. All of these efforts depend heavily on the interactive processing facility recently installed at RADC. This facility includes a flexible computing system as well as high performance graphics and allows the operator to both perform development functions as well as to simulate several applications functions for specific operational users. The Milestone Chart (Page 3.12-) outlines the computer graphics program.

The Air Force Global Weather Central (GWC) Interactive Graphics System effort is directed toward providing their analysts and forecasters with the ability to generate, edit, analyze and update various weather charts and forecasts in the interactive mode. This will increase the flexibility as well as decrease the cost of the various operations involved. The final system configuration will include numerous terminals and several forecast locations as well as a direct hook-up to the main GWC computer facility. This is a multi-phase, multi-year effort which will result in the automation of a significant portion of the GWC operation. This will be accomplished through the implementation of several mini-computer based graphics systems, as well as the development of specialized software and techniques to handle the particular characteristics of the environmental data. The effort is being pursued in support of MAC-DAR-L72-7 with full implementation scheduled for FY-78.

One of the prime considerations in applying computer graphics is that of economy. The specific approach that will reduce the per-terminal cost of multiple terminal graphics systems is the use of raster (or digital TV) techniques. Such a system not only has the potential for high performance at relatively low cost, but also offers the unique feature of being able to mix computer generated graphical data with pictorial information. The Raster Graphic Techniques effort is aimed at investigating the feasibility of performing the graphics functions using interactive TV displays. The goal is to provide performance equivalent to present selective address systems with a significant reduction in cost. The culmination of the program in FY-78 will be a Multi-Terminal Raster

Graphics System which demonstrates equivalent performance at much lower cost than previously available systems.

The Graphics Software Techniques area represents a group of related and continuing efforts aimed at decreasing the complexity and increasing the efficiency of graphics software and the means to generate it. To provide a testbed within which to work, a complete graphics software system has been configured to operate on our graphics facility. Among the areas which will be pursued in future development efforts are the investigation of graphic data base structures and their effect on processing efficiency, as well as the practicality and desirability of including a graphic extension within a higher order language, particularly the block structured declarative type.

The Graphics Modeling effort is aimed at the development of a modeling system to investigate the various parameters utilized in the design of interactive graphic systems. These include the use of satellite processors, distributed as opposed to concentrated data bases, the effect of response time on various interactive devices, and operator performance within such a system. The result will be more effective and efficient system designs while minimizing the overall cost.

3.12.3 RELATED EFFORTS:

(U) Numerous efforts which are related to this TPO are being pursued by other Air Force Laboratories and other agencies, both within DOD and in industry. Close technical liaison is being maintained on these efforts and any technical advances made will be applied to this TPO. Conversely, the results of efforts under this TPO are made available to other agencies for application to their programs. Following is a summary of the major efforts related to the products of this TPO.

3.12.3.1 (U) ASSOCIATIVE PROCESSOR COMPUTING SYSTEM - The Federal Aviation Agency is studying the use of an associative processor for air traffic control processing. A prototype STARAN was used for conflict prediction studies at the Knoxville, Tennessee test site. Further studies are being conducted with three contractors, each applying benchmark problems to their machine. The Defense

Mapping Agency (DMA) is leasing an associative processor to study its applicability to the post processing functions of cartography.

3.12.3.2 (U) DISTRIBUTED COMPUTATIONAL SYSTEM - The ARPANET designed and implemented by the Defense Advanced Research Projects Agency (ARPA) embodies many of the prerequisite features which will provide the basis for this advanced study. Specifically, it includes all the protocols which allow transfer of data and programs between computers and between peripherals and computers. The National Software Works program, sponsored by ARPA in conjunction with RADC TPO-11 "Software Sciences Technology", having the objective of giving a programmer access to many software development tools, will coordinate closely with and use many of the results of this program.

3.12.3.3 (U) CONFIGURABLE PROCESSOR FACILITY - Univac has a program under IR and D which will be used to design their next generation computer systems. They intend to build a microprogrammed multi-processor with an extremely flexible internal communications scheme to support the program. IBM, under internal corporate research, is designing a direct execution machine (a machine which directly executes higher order language code).

3.12.3.4 (U) TARGET RECOGNITION - The Army is applying pattern recognition techniques to speaker identification photograph analysis, shock trauma analysis and mine fusing applications. The Navy is using pattern recognition for speech recognition and for sonar target identification. NASA is doing work in identification of various earth resources using multi-spectral scanners. The Arms Control and Disarmament Agency is utilizing pattern recognition techniques to identify nuclear blast detonations.

It should be noted that the preceding work has been done in close cooperation with RADC. Some of the work was actually completed on the RADC system by other services after training and familiarization by RADC personnel. These agencies have either begun development or are considering development of a research tool patterned after the RADC On-Line Pattern Analysis and Recognition System (OLPARS); or are planning to continue using the RADC system.

3.12.3.5 (U) COMPUTER GRAPHICS SYSTEMS - There are numerous efforts in computer graphics being pursued both within DOD and in other government agencies. The Army is developing cartographic analysis techniques using interactive graphics while the Navy is synthesizing radar backgrounds and clutter to test radar signal processing systems. NASA has used computer graphics extensively for both computer aided design and flight simulation. More recently within the National Oceanographic and Atmospheric Administration computer graphics has been applied to the analysis of meteorological data particularly relative to severe weather conditions. ARPA has been one of the prime movers in the development of graphic techniques primarily through its university contracts.

3.12.4 REQUIREMENTS:

(U) There are numerous Air Force requirements for the technologies being developed under this TPO. High speed data processing techniques, software first techniques, automatic target identification techniques, and interactive graphics systems are being developed to satisfy the following requirements:

(J32770) 18-JUN-75 05:42;;;; Title: Author(s): Roberta J. Carrier/RJC; Sub-Collections: NIC; Clerk: RJC; Origin: <LORETO>TPO74.NLS;1, 20-AUG-74 16:13 DRL2 ;####;

1 3.11 TPO NO. 11 - SOFTWARE SCIENCES TECHNOLOGY 1

1a 3.11.1 GENERAL OBJECTIVES: 1a

1a1 (U) The general objectives of this TPO are to develop techniques to improve the reliability, reduce the cost, and increase the usefulness of computer systems to the Air Force. 1a1

1b 3.11.2 SPECIFIC GOALS AND TECHNICAL APPROACHES: 1b

1b1 (U) The Overview Chart (Page 3.11-2) illustrates the plan for meeting the general objectives. The objectives are a distillation of requirements of the systems listed on the right side of the chart. These systems are essential to any application of Air Force power in response to a directive from the President of the United States. In addition, data processing supports data reduction activities in practically all other systems. To support these systems, the five areas of effort in this TPO are oriented toward the major goals or products shown on the overview chart. Each of the areas of effort will be discussed in the order indicated on the overview chart. 1b1

1b2 3.11.2.1 (U) SOFTWARE QUALITY - The goals of this technical area are to provide the Air Force with capabilities to improve the quality (i.e., reliability, transferability, maintainability, efficiency) of its software while lowering the cost to attain that quality. This shall come about by the development of technologies for the quality control of computer Higher Order Languages (HOLs) and procedures for the generation of cost effective error-free software systems. 1b2

1b2a The goals of the Higher Order Languages (HOL) effort are: (1) to produce high quality compilers in an automated fashion, thereby cutting down on the cost and time necessary to build compilers, and (2) to develop the methodology necessary to control HOLs in order to eliminate certain problems with their use and thereby promote their use. 1b2a

1b2b The first goal is being achieved by capitalizing on state-of-the-art compiler building techniques and producing a compiler building tool called JOCIT, standing for JOVIAL Compiler Implementation Tool. The first JOCIT for implementing JOVIAL/J3 compilers was completed in November 1973, as can be seen from the milestone chart. The compiler was modified to meet some unique requirements of WWMCCS, was tested and accepted by WWMCCS users, and will become a WWMCCS standard compiler by September 1974. (Thus RADC was able to provide SAC and NORAD with an acceptable JOVIAL Compiler at a saving conservatively estimated to be $658,000). Again looking at the chart, it can be seen that a follow-on effort to develop a JOCIT for JOVIAL/J73 will be completed in FY-77. There are also plans underway to develop similar tools for FORTRAN and COBOL, and the totality of all four compiler tools would feed the Language Control Area which is discussed later. 1b2b

1b2c There are also efforts in this area which will provide the Air Force with the ability to evaluate its applications with respect to which HOL and/or type of compiler will meet its needs and also provide the ability to better specify the HOL or compiler. 1b2c

1b2d With respect to advancing the state-of-the-art of compilers, there is an ongoing effort to make extensible language compilers efficient enough for practical use, and a planned effort to implement a compiler for an "error resistant" language. 1b2d

1b2e The second goal of the Higher Order Language efforts is aimed at providing the Air Force with a measure of control over the HOLs it uses. The first effort undertaken was to develop a HOL called JOVIAL/J73 which is more responsive to Air Force needs and should thus increase HOL usage. The original specification of this HOL was completed in FY-73, and it is planned to modify the language in FY-75 and 76 to make it more "error resistant". 1b2e

1b2f The development of a compiler validation system is part of this effort. A compiler validator for JOVIAL/J3, called JCVS, has been developed and augmented to a degree where it is the most complete single test of a compiler in existence. The success of this tool has prompted the development of a similar system for JOVIAL/J73 compilers. An evaluation of compiler validators already in the field for COBOL and FORTRAN is also underway. A BASIC compiler validator is also under test in-house and a follow-on is planned to utilize modern "theorem-proving" techniques to build an "absolute" compiler validator. 1b2f

1b2g The problem of multiple interpretations of programming languages, arising from incomplete, ambiguous specifications, was attacked by the development of a system called Semantics Oriented Language (SEMANOL) which enables one to precisely specify and check out the syntax and semantics of a HOL. SEMANOL was applied to JOVIAL/J3 in FY-73 with satisfying results and will be utilized in FY-73 to "debug" the JOVIAL/J73 specification mentioned above. Future plans include the application of this system to other Air Force standard HOLs such as FORTRAN and COBOL. 1b2g

1b2h In order to collect proper data on HOLs used in the Air Force so that constructive changes can be made, statistics gathering packages for JOVIAL and BASIC are being developed, both in-house and contractually. These packages will utilize information available to HOL compilers to provide the data which was drastically lacking in the past when HOL or compiler updates were attempted. 1b2h

1b2i Other work in this area includes studies into HOL requirements of specific Air Force systems such as DAIS (Digital Avionics Information Systems) and the comparison of all HOLs in use by DOD. 1b2i

1b2j The last product on the milestone chart, in FY-80, represents the gathering of the tools and technology developed from the Compiler Technology and Language Control area into neat packages which will give the Air Force complete control over all standard HOLs it plans to use now and in the foreseeable future, (i.e., FORTRAN, COBOL, JOVIAL/J3 and JOVIAL/J73.) To help in this control, the design of a Language Control Facility and the implementation of a reporting system will begin in FY-75. 1b2j

1b2k The goals of the Software Reliability efforts are to investigate and develop techniques for increasing the reliability of complex system software. 1b2k

1b2l Initial design of a centralized data facility on software error, cost and productivity data was initiated in FY-74. An effort was initiated to study existing methods of detecting and evaluating software failures during testing and operational phases of large Command and Control Software Systems. Efforts were also initiated to study the nature of software reliability modeling, software errors, their classification and number, their removal during testing and correction, the prediction of their occurrence, and techniques for writing low error content software. A prototype Software Implementation Monitor (SIMON) was also designed in FY-74 and is currently under development. 1b2l

1b3 3.11.2.2 (U) SOFTWARE ENGINEERING - The goals of this Technical Area are to develop programming aids and tools required by programmers, test engineers and maintenance personnel and to develop the means to make these tools readily available for DOD software developments across the country. 1b3

1b3a The goals of the Programmer Tools effort are to develop software tools which will assist the writers or programmers of software, and support test engineers and maintenance personnel. These tools are generally in the form of computer programs, support software and other automated aids and provide help in the generation of software. A multitude of tools have been developed and are currently available. Unfortunately, these tools are usually on a specific machine and/or for a specific language. A vehicle for making these tools available despite this drawback is the National Software Works (NSW) which is discussed later. A number of tools which have great potential for improving software reliability and productivity are proposed or are currently in development, e.g., structural complexity analyzers and automatic error data collectors. Analyzers which determine the extent of compliance to programming conventions, and analyzers for searching out coding blunders are planned for development in FY-75 and FY-76. 1b3a

1b3b Emphasis in the area of software test tools will be placed on the development of new and more powerful tools which support extensive analysis of logical paths in software as a function of input data sets. The systems also provide valuable statistical records of tests completed, logical paths tested and not tested, and assist in the generation of new test cases and timing/code optimization. 1b3b

1b3c New tools are also being developed such as block analyzers for determining scope of variables, and loop analyzers for determining potentially singular point conditions. 1b3c

1b3d Other approaches to testing software involve methods for proving software formally correct and include inductive proofs on programming assertions, semi-automatic theorem provers, and exhaustive test case analysis. 1b3d

1b3e Because of the diversity of existing tools for testing, and writing software, a significant problem to be faced in quality software production is that of cataloging and classifying the different types of tools available and of making these tools available to Air Force and other government users. This cataloging will include an analysis of each software tool to determine languages or operating system dependencies, known errors or deficiencies, and functional requirements and other peculiarities of the package that may impact intended users of the tools. Also included in the analysis will be recommendations for improving the general utility of each tool for wider usage, and the possibility of moving the tools to other environments and translating the packages to other programming languages. 1b3e

1b3f The goals of the Modern Programming Practices area are to undertake the development of techniques for formalizing the design of software. 1b3f

1b3g Traditionally, software design has been based upon natural language narratives, flow charts and decision tables. New approaches currently being (or planned to be) investigated involve the use of the HIPO technique (hierarchy-input-process-output), transition diagrams, levels of design abstraction, pseudo languages and special design languages, such as the University of Michigan's PSL (Problem Statement Language), National Cash Register's ADS (Accurately Defined System), and IBM's PDL (Problem Design Language). Fully automatic translation from design to programming language is considered a design goal in many of the above systems. 1b3g

1b3h Other techniques being considered for the designer include systems for testing the completeness of narrative type software specifications for consistency checking, and for performing termination checks on software designs. Other efforts being undertaken include the development of structured programming preprocessors for JOVIAL, COBOL, and FORTRAN, and implementation of programming support libraries for the above languages. 1b3h

1b3i A detailed set of guidelines will also be completed in FY-75 that will serve to transfer present technology in structured programming (SP), top-down programming, chief programmer teams (CPTs) and programming support libraries (PSLs) to the Air Force for further application. Areas to be investigated include the development of SP language standards for COBOL, FORTRAN, JOVIAL J3 and J73, analysis of data structuring methods, and development of requirements for a CPT and PSL. Other aspects of software quality architecture and software quality engineering will also be explored including the metrics of software quality. 1b3i

1b3j The goal of the National Software Works effort is to improve the productivity of DOD software development by making available to programmers and managers, in a uniform fashion, those types of software tools in widespread use in the research community. 1b3j

1b3k Two equally important tasks confront the R & D laboratory: the development of tools and the exporting of tools. Once a particular set of software tools is developed, a means must be available to distribute these tools to various users. Often, this task is hampered because of the transferability problem - the problem of getting programs developed at RADC on one computer out into a user community of different computer types. Because software can be transmitted accurately over digital communication links, a promising solution is to make these software tools available through the ARPA sponsored National Software Works (NSW). 1b3k

1b3l The NSW is directed toward the development of the necessary interface programs on the ARPANET to allow users access to an integrated set of tools and programming and/or management aids which might exist at different installations or sites throughout the country. A user at one site would have a standard method of using a tool at a site thousands of miles away. The NSW would provide a framework for tool encapsulation, centralized accounting, centralized file management, and front-end processing. 1b3l

1b3m NSW service will begin in FY-76 with the connection of the Data Services Design Center as a user bearing host. Later, RADC will connect its MULTICS and WWMCCS computer to the ARPANET in order to make available RADC developed tools. Thus the often sought links between the laboratory and the user will be achieved. Minimal modifications to the operating systems will be made to convert the RADC computers into tool bearing hosts. 1b3m

1b3n NSW and the ARPANET will provide an environment whereby the producers of software tools can easily interact with the user of the tools. Files and programs can be transferred between sites with ease. Information can quickly pass from the tool producer to the tool user and back again in much less time than before. Where the laboratories before were isolated centers of research, now a close interaction with users is possible. 1b3n

1b3o The development of a quality software production and analysis center requires just such a close interaction between users and producers of tools. A software reliability center can be implemented which maintains data on the quality of software produced by vendors. Information on software running in the field can be returned to the center for updating reliability files. Samples of software can be transferred to the center through NSW and analyzed for quality. Statistics can be collected and retained for future software contractor qualification. In this way high standards for software vendor selection can be maintained. This would directly aid a USAF acquisition which has software components. 1b3o

1b4 3.11.2.3 (U) SYSTEM SOFTWARE - The goals of this Technical Area are to support Air Force WWMCCS sites, enhance the capability of WWMCCS, develop for users the ability to manipulate large data bases and to develop techniques for providing multi-level security. 1b4

1b4a The goal of the Security Technology effort is to develop the ability to share EDP systems and the information therein with the assurance that classified information stored and processed will receive appropriate protection. 1b4a

1b4b The specific goals and products of this program are organized into three categories: 1b4b

1b4b1 The development of an abstract mathematical model of computer security design requirements compatible with the Department of Defense security system for protection of classified information. Further, the development of an explicit methodology for application of these models to the design and certification of specific systems. 1b4b1

1b4b2 The development of a prototype secure computer system which will provide the multi-level information processing capablity described under General Objective. 1b4b2

1b4b3 The application of the computer security techniques developed to specific Air Force requirements. In particular, technology transfer support will be provided to current ADP system developments which presently cannot satisfy economic constraints nor required operational capabilities due to the current lack of valid computer security controls implemented in the hardware and software. 1b4b3

1b4c The approach being taken by ESD for developing a secure computer system begins with a formal definition (i.e., model) of secure operation. This model defines the operations necessary for the utility as well as the security of a computer system. A hardware/software security kernel, totally responsible for the system's security, then takes place as a direct implementation of the security model. ESD has completed two different types of models for computer security and has implemented one of them on a PDP-11/45 minicomputer to provide a feasibility demonstration. The development of a secure general purpose large scale computer system is being undertaken based on the methodology developed during the feasibility demonstration. This development, using the security kernel approach, will use an appropriate target computing system whose utility has been previously established. In addition, development of audit, surveillance, and secure DMS subsystems will be undertaken to provide application aids for a secure computer system. Finally, to facilitate use of secure computer systems, more efficient, less expensive terminal and communication security equipment designs will be developed. 1b4c

1b4d Software Executive Services - The goals of the Software Executive Services effort are: to provide requirement analysis, quality assurance, testing support and design recommendations for the Air Force WWMCCS community, and to provide system software support to operational users requiring capabilities to manipulate large data bases. 1b4d

1b4e RADC, at the request of Hq USAF/ACD, completed a study of the Extended Instruction Set (EIS) for the Honeywell 6000 series computers. The results of the study had a major impact on the decision by JTSA (Joint Technical Support Activity) to upgrade all the WWMCCS computers to include EIS. 1b4e

1b4f In addition, RADC, using the RADC Honeywell 635 computer, studied the feasibility of using Honeywell 600 computers for the major command update program and as backup for the H/6000 WWMCCS. Results of the RADC effort had a major effect on the Hq USAF/ACD decision not to use the H/600 computers. 1b4f

1b4g RADC developed a technical plan for the Air Force Data System Design Center (AFDSDC) to assist that oganization in performing their function as the Air Force focal point for reporting WWMCCS deficiencies to JTSA. The plan is currently being followed and RADC has become the prime source for technical assistance to AFDSDC. A letter of agreement has been initiated defining the roles of RADC and AFDSDC in supporting the Air Force WWMCCS sites. 1b4g

1b4h RADC, in cooperation with ESD/MITRE and JTSA has performed field tests on the Honeywell developmental Network Processing System (NPS) to test its functional capabilities and to determine its utility/applicability to the WWMCCS community. The test reports are being prepared. Analysis of results point out that current NPS implementation decreases system throughput in comparison with existing software. 1b4h

1b4i The technical efforts described above, active involvement with WWMCCS users through requirements conferences and documented needs plus explicit direction in PMD R-P4010(1)/63728F Technical Support to Air Force HIS 6000 Sites have led to a two pronged development program. A number of tasks are proposed which address development of fixes to current pressing problems which must be solved in the near term to allow users to perform their missions. A longer term solution, running concurrently with the fixes, is proposed to attack the user problems caused by deficiencies inherent in the design of the WWMCCS operating system software. 1b4i

1b4j The major thrust of this program is to improve the responsiveness of the WWMCCS system software to meet the stated requirements of its users. The Restart/Recovery task addresses the requirements from SAC and NCRAD to have the system 95 to 100% available. This is currently accomplished with costly redundancy. The short term approach addresses procedures which may be used in conjunction with the existing software/hardware configuration to facilitate recovery of data bases and decrease the time to restart the system and return the users to the point of processing just before system failure. The goal of these procedures is always to attain zero loss of information and minimal system downtime. 1b4j

1b4k Real Time/Priority Processing - SAC has a documented requirement for response times of less than two seconds during crisis situations. TAC has stated they require less than ten second response for 50% of their application jobs. These requirements cannot be met with the current WWMCCS. Our approach is to modify the Honeywell provided TPE (Transaction Processing Executive) to achieve real time processing. We are currently under contract with Honeywell to provide these modifications. Design is almost completed, flow charting will start immediately. After completion of the models an effort is planned to apply this strategy to real time transaction oriented applications. 1b4k

1b4l Front End Processing - Plans are to continue to test new releases of the Honeywell NPS (Network Processing System). We will play a major role in developing a firm NPS specification which will eventually become a standard package in WWMCCS. 1b4l

1b4m File System - The current response time problem with data retrieval under WWMCCS is a direct result of the limitations of the WWDMS and GCOS file system. Using existing tools previously developed by RADC (data access systems, flexible indexing systems and software monitors), we plan to model various approaches for handling the retrieval problem in order to lead us to intelligent design changes critically needed for WWMCCS. 1b4m

1b4n System Architecture - Up to now we have been talking about plans and efforts which address enhancing specific components of WWMCCS. They represent modifications or patches for near time solutions. We also plan to conduct efforts which will address the enhancements of the total WWMCCS in terms of reliability and responsiveness. These efforts will be based on arriving at a modularized operating system which will replace GCOS. The pay off would include: 1b4n

1b4o Configuration according to specific requirements, thereby increasing responsiveness and reducing overhead, and clearer code which will reduce software errors and increase reliability. The concept of Virtual Machine Monitor (VMM) is one of the approaches we plan to pursue to arrive at modularity. We plan to initiate a contract with the Harvard Computation Laboratory to study the fundamentals of VMM and also plan to initiate an effort to study the applications of VMM to the Honeywell 6000 computer. In this application, an attempt will be made to run GCOS and MULTICS simultaneously on the H/6000 computer. 1b4o

1b4p In addition to supporting WWMCCS, RADC is also supporting HQ AFCS and the Defense Mapping Agency in providing aids to their large file problems. RADC has provided HQ AFCS an in-house developed on-line retrieval system operating on the RADC/H635 computer to support their operational needs. In FY-75 it is planned to augment the current capability with additional file manipulating functions. The Defense Mapping Agency has requested that RADC examine their large file processing needs, with the goal of designing a management information system for their cartographic data base. In FY-75 an initial study of this problem is planned. 1b4p

1c 3.11.3 RELATED EFFORTS: 1c

1c1 (U) The following efforts are related to work being pursued under this TPO. In general, compiler-compiler efforts being pursued by industry involve "many-to-many" language-to-computer configurations. This is different from RADC's "one-too-many" approach in JOCIT, which it is felt should produce higher quality compilers. In addition, the JOCIT effort is the only known effort producing compilers for the JOVIAL language. 1c1

1c2 Automatic Validation Systems (AVS) efforts under investigation by other organizations are principally concerned with the FORTRAN language, hence no developments are being pursued along this line other than transfer of a FORTRAN test tool to RADC. Development of an AVS type capability for JOVIAL is required. 1c2

1c3 In the area of data management software, industry is now producing generalized data management software hence the emphasis for this program is shifting from large scale developments to tools for specifying, selecting and tuning generalized data management software. In security, ARPA sponsored work at MIT on Multics will be used directly in the development of a long range solution to the security problem. In the tools for knowledge workers area, the ARPA sponsored research at Stanford Research Institute is being exploited directly and other related research is being followed closely. There is no other major activity in this area where a subset of an organization is attempting to systematically exploit sophisticated on-line computer tools. 1c3

1c4 All other software efforts are being considered either complementary to work being pursued at RADC, or are serving as a baseline upon which further advancements are being made. None of the efforts below are considered duplicative in any way. 1c4

1c5 SOFTWARE QUALITY 1c5

1c6 NASA presently has a contract with McDonnell Douglas (Contract Number NASA-27202) to design a compiler-compiler capable of producing compilers for all HOLs which NASA uses, or plans to use, such as FORTRAN, SPL, CLASP, etc. 1c6

1c7 NELC has a contract with Intermetrics Corporation (Contract Number N00123-73-C-1177) to design a HOL for the AADC computer. This effort is under Project W3150, Program Element 63202N. 1c7

1c8 The U.S. Army Electronics Command is developing a Compiler Generation Tool for TACPOL, a PL-1 Command and Control Subset. This work is being performed in-house under System Software Program Element 627703, Program Element 15662703A327, Task Element 03, Work Element 36108. 1c8

1c9 IBM Federal Systems Division, Owego, is working on a compiler-compiler to handle several DOD HOLs. 1c9

1c10 Boeing Corporation is performing an analysis of present HOLs for B-1 follow-on implementations. At the present time, JOVIAL/J73 is one of the strongest contenders. 1c10

1c11 Univac of Minneapolis is using IR&D funds to develop translators between the Command and Control HOLs in use by DOD, namely, JOVIAL, CMS-2, and TACPOL. 1c11

1c12 AFAL has a program entitled the Digital Avionics Information System (DAIS) under the TPO-S1 which has chosen JOVIAL/J73 as the HOL in which its mission software will be written. 1c12

1c13 NASA/MSC has implemented an AVS for FORTRAN programs written for the Univac 1108 as part of the Mission Trajectory Control Program and the Skylab Activities. 1c13

1c14 Software Reliability studies at Carnegie-Mellon University, University of Wisconsin and MIT in Common Base Languages (Dennis) and NSF sponsored work at SRI and University of California at Irvine are also ongoing government sponsored programs. 1c14

1c15 Air Force sponsored work on the Attack Assessment Program for SAMSO includes provisions for collection and analysis of software error data. 1c15

1c16 U.S. Army and Navy sponsored work at PIB (Shooman) involves software error collection/analysis and procedures for software modeling and reliability prediction. 1c16

1c17 SOFTWARE ENGINEERING 1c17

1c18 The Air Force is currently AVS testing FORTRAN programs on the IBM 360/370 and Assembly Code on the IBM 7090 written as part of the Minuteman Program. 1c18

1c19 The Army is investigating the problem of verifying Safeguard Software at Huntsville, Alabama. 1c19

1c20 The Navy is investigating the software verification problem in support of system software activities at NEL. 1c20

1c21 Projects employing Structured Programming and Chief Programmer Teams and Top Down Programming include government sponsored programs such as: Safeguard (PAR Program), AWACS/JOVIAL Support, RTCC/Skylab Real Time Computing Complex); Systems 7, 370/EMS (Energy Management System), NMCS and NIPS System 360FFS (National Intelligence Processing System 360 Formatted File System). 1c21

1c22 SYSTEM SOFTWARE 1c22

1c23 ESD has been working with the computer industry and has participated in computer security test programs. A description of these efforts is presented in ESD-TR-73-51, "Computer Security Technology Planning Study." 1c23

1c24 Several industrial IR&D efforts concerning computer security have taken place. These include the Data Security Study performed by IBM's Systems Development Division, efforts by the Data Systems Operations Division of Honeywell Information System Inc., and projects on computer security at System Development Corporation and TRW Systems Division. These efforts will not meet Air Force security requirements without the guidance to be provided by this Technology Planning Objective. 1c24

1c25 ARPA has supported a development program under Project MAC conducted at Massachusetts Institute of Technology to simplify the security related aspects of Multics. This program is now jointly sponsored with the Air Force and is the basis for the large scale, general purpose secure prototype system development. 1c25

1c26 NSA has established a computer security division that is investigating a number of computer-related communications security problems. NSA is expected to take a leading role in the communications security portions of this program. 1c26

1c27 The Joint Technical Support Activity is currently involved in studying ADP multi-level security capabilities, data management system multi-level features, and ADP network security. The JTSA program is being fully coordinated with this program to prevent overlap. Some, but not all, of the security problems confronting the Air Force WWMCCS community are being addressed in this effort. 1c27

1c28 SAC, through the 436M SPO, has Computer Sciences Corporation developing an on-line interactive control program called SONIC. Initial investigations by RADC and the Air Force Data Systems Design Center reveal that its transferability to other WWMCCS users is minimal. 1c28

1c29 RADC/IR has a front end processing development program called TOSS (Terminal Oriented Software System). It is currently being considered as the prime candidate for intelligence networking applications. 1c29

1c30 The SAC PACER program, being directed by RADC/IR, has a continuing effort in restart/recovery, with major orientation to data base integrity and recovery. 1c30

1c31 NORAD, through the 427M SPO, has developed a Real Time Monitor (RTM) to handle their interactive processing requirements. MAC has developed a real time operating system (RTOS) under WWMCCS operating system. The Joint Technical Support Activity has a continuing effort in improving the Worlwide Data Management System (WWDMS) to meet the WWMCCS target requirements. 1c31

1d 3.11.4 REQUIREMENTS: 1d

1e In addition to the overall objectives indicated on the overview chart, this TPO is responsive to the following requirements: 1e

1f Requirement Identification Number	Title	TPO Applicability	
			1f

1g SOFTWARE QUALITY (HIGHER ORDER LANGUAGES) 1g

1h CCIP-85 AF Command & Control
Essential
Information Processing
1980's 1h

1i SADPR-85 AF ADP Requirements
Essential
1980's 1i

1j SOFTWARE QUALITY (SOFTWARE RELIABILITY) 1j

1k CCIP-85 AF Command & Control
Essential
Information Processing
1980's 1k

1l SADPR-85 AF ADP Requirements
Essential
1980's 1l

1m SOFTWARE ENGINEERING (PROGRAMMING TOOLS) 1m

1n AFSC Program Technical Support for
Essential
Directive AFDSC Multics
1992/01-1-73/27 1n

1o CCIP-85 AF Command & Control
Essential
Information Processing
1980's 1o

1p SADPR-85 AF ADP Requirements
Essential
1980's 1p

1q SOFTWARE ENGINEERING 1q

1r CCIP-85 AF Command & Control
Essential
Information Processing
1980's 1r

1s SADPR-85 AF ADP Requirements
Essential
1980's 1s

1t SYSTEM SOFTWARE (SOFTWARE EXECUTIVE SERVICES) 1t

1u USAF Development Command, Control & Data
Essential
Directive 79 Systems Software 1u

1v AFSC Program MAC Integrated Mgt
Significant
Directive System (MACCCIMS)
415L-2-73-51 1v

1w AFSC Program Technical Support for
Essential
Directive AFDSC Multics
1992/01-1-73/27 1w

1x AFSC Program Acquisition of Multics
Essential
Directive for AFDSC
1992/02-1-73-28 1x

1y SATIN IV PMP SAC Automated Total
Essential
R-P4020-(1) Information Network
(SATIN IV) 1y

1z CCIP-85 AF Command & Control
Essential
Information Processing
1980's 1z

1a@ SADPR-85 AF ADP Requirements
Essential
1980's 1a@

1aa STALOG Study of Automation of
Essential
Logistics 1aa

1ab AFSCNET AFSC Computer Network
Significant 1ab

1ac PMD R-P4010(1) Technical Support to
Essential
AF HIS 6000 sites 1ac

1ad SYSTEM SOFTWARE (SECURITY) 1ad

1ae TN-ESD-24-72-17 Security Controls in Multi-Users Computer Systems Essential 1ae

1af SAC-ROC-74-1 Multi-level Security Essential 1af

1ag USAF Development Directive 79 Command, Control & Data Systems Software Essential 1ag

1ah ESD-TR-73-51 Computer Security Requirements of Several Major AF Commands Essential 1ah

1ai PMD-R-P2145(2) 33401F Secure Telecommunications Terminals Significant 1ai

1aj CCIP-85 AF Command & Control Information Processing 1980's Essential 1aj

1ak 3.11.5 RESOURCES: 1ak

1ak1 (U) The funding and manpower required to support this TPO is listed on (Page 3.11-24). No other major support requirements have been identified. 1ak1

1al 3.11.6 TPO FOCAL POINT: 1al

1al1 1al1
1al2 FRANK J. TOMAINI 1al2
1al3 Rome Air Development Center (ISI) 1al3
1al4 Griffiss AFB NY 13441 1al4
1al5 Autovon 587-7507 1al5

(J32771) 18-JUN-75 05:44;;;; Title: Author(s): Roberta J. Carrier/RJC; Sub-Collections: NIC; Clerk: RJC; Origin: <LAFORGE>TPO11.NLS;1, 3-SEP-74 10:47 ELF ;####;

Response to Teleconferencing questionnaire.

This is only a personal response. Let me know your feelings by sending pertinent mesages messages to the teleconferncing group ident. Thanks.Glenn, sorry I am slow in responding.####;

Response to Teleconferencing questionnaire.

1 FROM: Lieberman (RLL) 1

2 SUBJECT: Response to Teleconferencing questionnaire 2

3 Q1. Research to date indicates that there have been at least 8-10 separate attempts around the country to develop and implement computer conferencing packages. These include FORUM, PLANET, CONFER (PLATO IV) PARTY LINE, CONFERENCE/RIMS (Turoff), EMISSARY, ORACLE, TCTALK (Calvin) and General Conferencing System (GCS). (See attachment for background and details). Have you ever used any of these systems or any other teleconferencing system? 3

4 Q2. If yes, which one or ones? 4

5 Q3. How much experience have you had with such systems, i.e., number of times used, duration of use, etc.? Approximately when (i.e., calendar dates)? How many others participated? 5

5a A1, 2, 3: Yes. FORUM, very little about 18 months ago but not in a real conference; DOUG off and on only as a test. 5a

6 Q4. Are you convinced that teleconferencing should be available to NLS users on Office-1, -2, etc.? If so, could you tell us briefly why, or why not? 6

6a The addition of teleconferencing to the Office-n environment would be extremely useful. I believe this will be due to the similarity of the sort of things teleconferencing and NLS are striving for. 6a

6b The same kind of people that have accepted NLS are also most likily to accept teleconferencing. 6b

6c In fact, I believe that a disadvantage might result from including a teleconferencing system within the NLS world, that is, the increase use of the computer due to conferences. Also this might increase the desire of having more people on at the same time from one group. 6c

6d This might be very hard unless the pie slice scheduler is soon to be activate under 1.33. 6d

6d1 There might be some contractual problems but I think this could be solved by an agreement by all. 6d1

7 Q5. What do you think you would use it for? 7

7a As a partial substitute for the sndmsg, journal, and link facilities. 7a

7a1 Linking will remain as the unplanned computer "telephoning". 7a1

7b As a training aid. 7b

7b1 Training courses could be given on NLS as a conference. 7b1

7c To get higher level people on line. 7c

7c1 This might not be a good idea. 7c1

7d To broaden the communication features of NLS by providing multi-person real time and asynchronous interaction. 7d

8 Q6. How would this use be different from mailbox capabilities such as message transfer in TENEX and the NLS journal system that are already implemented? 8

8a The flavor of real time interaction is not present in the journal or sndmsg. 8a

8b More importantly the sequence of the conversation on a particular subject will be clearer. 8b

8c Also, there is the potential that a conferencing system will be easier to learn and use than even a small, simple subset of NLS. 8c

9 Q7. Based on your experience, if any, with computer conferencing, which system(s) that you know of best represent what you think a teleconferencing system should be. 9

9a Unknown to me. Not enough experience. 9a

10 Q8. From the brief review of teleconferencing attached to this document, can you identify features that you think would be especially useful in an ARC teleconferencing package? Any specific features not especially useful? 10

10a Permanent record kept of conference sessions (to be integrated with Journal I guess) 10a

10b I would like to see both the 'speaker floor' concept and the totally asychronous mode combined. 10b

10b1 Thus only one person could have the floor and everyone vies for it in perhaps some queue. Listeners could have the option of listening only to the 'speaker' or to both the speaker and the background comments. 10b1

10c Participants can be asynchronously conferring. 10c

10d Reading of previous transcripts in a variety of ways (hopefully as an NLS file). 10d

10e Initially I think DELPHI capabilities will not be needed by NLS users. 10e

10f Also I believe that searching past transcripts by means of author, keywords, etc. is not needed (hopefully the files will be in NLS and can be searched there). 10f

10f1 More to the point searching should be left to NLS and not as part of a conferencing system. Any particular searching that conferencing would like to have should be some way incorporated into NLS or written as a separate retrieval subsystem. 10f1

10g Private notes might be excluded from conferencing subsystem since they are in effect part of the SNDMSG or JOURNAL subsystem. 10g

10g1 This assumes that a participant of the conference can go to NLS while in a conference and return to it. It also hopes that an immediate feature is integrated into the journal system so that both permanance and unique numbering can be maintained. Perhaps same automated way of relating a journal item to a particular conference is also necessary. 10g1

10h The feature of supporting spinoff conferences seems minor (in that that can be easily done). 10h

10i Voting is also an easliy accomplished facility. 10i

10j Most of the retrieval needs and reading facilities can be or could be satisfied by NLS. 10j

11 Q9. Considering other possible improvements, embellishments, etc., that ARC development could be spending its time on over the next 12 months, how would you rank the addition of teleconferencing? 11

11a Urgent 11a

11a1 Don't think so in any case. 11a1

11b High 11b

11b1 Doubt it, there does not seem to be higher enough need or interest in making this a high priority item. Yes, there is considerable interest in getting a conferencing system but none of it appears very high. 11b1

11b2 Maybe if a simple-minded system was added. 11b2

11b2a The problem here is that a simple system often becomes the vehicle for whatever follows and that simple often is not what it becomes even at the first crack. 11b2a

11c Medium 11c

11c1 Probably the proper level for the next 12 months. 11c1

11d Low 11d

12 Q10. Would you favor getting an existing teleconferencing package onto Office-1, -2, etc. as soon as possible (if possible), or the development of a package specifically written for KWAC, by ARC research and development programmers and staff? 12

12a This depends on the community needs. No doubt whatever system is integrated into NLS it should be modified so that referencing to NLS statements con be included. The transcript should also be in NLS files. Thus, it seems that some critical work will have to be done for existing system and this would decrease the suitability of other systems. However, I would think some other system would be a good starting point. 12a

12b If what the needs are does not include some of the more complex features of teleconferencing than it might be the same cost to start from scratch within the NLS environment. 12b

12c Also the whole concept of what teleconferencing is must be examined for its proper connection to the journal system as well as the editing features. 12c

13 Q11. If the latter (Q10), would your organization be willing and in a position to contribute financially toward the development of an ARC-based (or NLS-based) package? 13

13a Norton's offer of 3 man months is still good. 13a

14 Q12. Do you have any other comments, questions, gentle cajolings or whatever to add to the discussion? 14

14a priorities must be set and once set they must be discussed with ARC. Also, the KWAC should quickly prepare a list of requirements they would like to see in the highest priority developments. The funding area must also be discussed and established. 14a

15 Thanks very much! 15

Response to Teleconferencing questionnaire.

(J32772) 18-JUN-75 07:20;;;; Title: Author(s): Inez M. Mattiuz/IMM;
Distribution: /GCE([INFO-ONLY]) ; Clerk: IMM; Origin:
<GJOURNAL>25866.NLS;1, 13-MAY-75 17:06 XXX ;;; Title: Author(s):
Robert N. Lieberman/RLL; Distribution: /GAS2([ACTION]) TELECON([
INFO-ONLY]) JCN([INFO-ONLY]) JHB([INFO-ONLY]) ;
Sub-Collections: SRI-ARC TELECON BELL-CANADA; Clerk: RLL;
Origin: < LIEBERMAN, CONF.NLS;9, >, 13-MAY-75 17:23 RLL ;;;;####;

1 HELLO...This represents the first cut at initiating a constructive dialog among KWAC'ers who expressed an interest in Teleconferencing. What follows is a general discussion of conferencing developed by RA3Y Panko to serve as background, and a questionaire that is intended to prompt relevant feedback from you. 1

2 Please use the Journal system (Sendmail) to send feedback to GAS2. Until the obvious coordinator(s) emerge from KWAC I shall act as the focal point for TELECON. Suggestions as to how and where to proceed from here are eagerly sought as well...Thanks Glenn. 2

3 3

4 A. INTRODUCTION 4

4a In the Autumn of 1974, some 29 scientists in the U.S., Canada, and England participated in a conference on how communication might someday substitute for travel. True to the spirit of their topic, the conferees did not all travel to some distant city; they conversed solely by computer. They used a computer teleconferencing program, which recorded public comments in a permanent record and made them available to all conferees. 4a

4b Computer conferencing programs are now available on national time-sharing networks in the U.S. and Canada. Although the government research funding that spurred the development of computer conferencing in the early 1970's has been greatly reduced, commercial companies are continuing the development of teleconferencing tools. 4b

4c There are now enough vendors of teleconferencing programs that most organizations can experiment with computer conferencing among their own members. Such experiments, however, must be evaluated carefully. While several available systems are quite useable and sophisticated, future systems will be even more powerful, and probably simpler to learn. In addition, the economic picture for teleconferencing is still fluid. 4c

4d The main virtue of computer conferencing, as opposed to live conferences, audio teleconferences, and video conferences is that not everybody need be present at all times. The computer-generated conference record allows participants to read earlier comments and respond to them. Computer conferences may last days, even months, and may include thousands of entries. 4d

4e Because man/computer conferences have long duration, it may not be entirely proper to call them "conferences" at all. A conference has traditionally been viewed as an episode in an organization's general communication process. Computer conferencing, with its unlimited duration, tends to blur the distinction between conferencing and general organizational communications. 4e

5 B. HISTORY 5

5a Computer-mediated communication began in the late 1960's with the emergence of time-sharing. Online users on many systems could "link" terminals, engaging in real time, albeit primative, conferences. In some early time-sharing systems, it was also possible to type messages to offline users; these messages would be delivered later, when the user next logged on. This latter "mail box" facility is now available on virtually all time-sharing computers and computer networks. Users are able to retrieve or file past mail by date, author, title, keyword, and other parameters. 5a

5b Mail box services, however, do not lie along the direct evolutionary line that led to computer teleconferencing. The real impetus to computer conferencing came from Delphi conferences. In Delphi, experts respond to a set of propositions, ranking the desirability or likelihood of difference propositions. Each respondent must fill out a questionnaire, and interactive computer systems seemed to offer a way to simplify and speedup questionnaire-taking and analysis. 5b

5c Two notable Delphi systems were created around 1970. The first was ORACLE at Northwestern, designer James Schuyler and Robert Johansen. The other was Delphi Conference at the Office of Emergency Preparedness (OEP), designed by Murray Turoff. 5c

5d In both of these systems, the intent was to gather expert opinion. And in both cases, the Delphi structuring was found to be too rigid. About 1972 both ORACLE and OEP began to develop free-form systems. 5d

5e The OEP development was spurred by the wage-price freeze in 1973. The OEP developed a full computer conferencing system called CONFERENCE. In subsequent crises, the OEP and its successor, the Office of Prreparedness in the General Services Administration, integrated computer conferencing and a management information system. This integrated system was called RIMS (Resource Interruption Monitoring System). 5e

5f CONFERENCE and RIMS, although useable systems, were not sophisticated nor highly user-prodified. Neither developed extensively after their creation. Two successors of CONFERENCE, however, are notable. One is the General Conferencing System (GCS), developed by memo from Turner and available on a Canadian time-sharing system, the I. P. Sharpe network. The second is Computer-Mediated Interaction (CMI), which is being developed by Bell Canada, and its subsidiary, the Bell Northern Research Laboratories. These systems, although heavily influenced by CONFERENCE and Murray Turoff, are essentially new designs. 5f

5g The ORACLE system has continued development to the present. A similar system, CONFER, is implemented on PLATO-IV. Neither has seen extensive use. 5g

5h The most sophisticated system widely available is FORUM (which is offered commercially as PLANET). Development began in 1971, with substantial funding from NSF and the Advanced Research Projects Agency. The latest version of Forum (FORUM-6) is quite sophisticated and allows the conference chair person to retrospectively analyze use patterns. Over a dozen conferees have been held with FORUM/PLANET. 5h

5i Teleconferencing appears to be proliferating now. Many computer systems are installing conferencing programs. Almost all of these in-house systems are quite primative, but they may serve an organization's basic needs. 5i

5j We have said above that computer conferencing is very difficult to separate from general computer-mediated communication. Several general computer communication systems now exist. It seems reasonable to expect that computer conferencing and general communication systems will eventually merge. What a joint system would be like is difficult to assess. 5j

5k One general communication system is NLS (On-Line System) developed at the Augmentation Research Center at Stanford Research Institute. NLS allows users to compose documents, messages or numerical files and send them to other NLS users through "Journal Mail". The Journal catalogues each piece of mail for later retrieval and sends citations to the receivers. In essence, the Journal both delivers and files messages. In the future, users may be able to read and send messages to "sets" of entries that will resemble a teleconferencing record. The sets capability is not yet available, but it illustrates a possible direction for development. NLS is presently used by about 200 people in 12 organizations in the U.S. and Canada. It is used for document preparation, mail transmission, and various other purposes. The number of users and organizations is expected to double within a year. 5k

6 C. CAPABILITIES OF TELECONFERENCING SYSTEMS 6

6a Computer conferencing systems vary considerably in general features, degree of sophistication, and limitations. This section gives an overview of what general capabilities and limitations an organization can anticipate in a teleconferencing system. 6a

6b 1. General Features 6b

6b1 In a face-to-face conference, a group of people discuss topics of mutual interest and make decisions. A computer conference must achieve the same ends. In general, computer conferences parallel the features of live or face-to-face conferences, but there are important differences between the two conferencing modes. 6b1

6b2 In a live conference, people make their remarks verbally to those around a table. In computer conferences, comments must be typed in, and they are made to a conference "record". The record is usually a linear list of comments; new comments are appended to the end of the list as they are received. All conferees may read the record. 6b2

6b3 Since the record is permanent, computer conferees need not be present when entries are made. Yet experience has shown that interactions frequently do occur among conferees who are online simultaneously, so it important for the record to be updated as soon as new entries occur. 6b3

6b4 In a live conference, private notes are also passed among conferees. In computer conferences, private messages can also be sent "off the record" to other participants. 6b4

6b5 Position papers, graphics, and the like are frequently passed among participants in live conferences. Some teleconferencing systems, but not all, allow non-message entries to be placed in the record and to be referenced in subsequent comments. 6b5

6b6 Voting occurs with some frequency in live conferences. Many teleconferencing systems also allow voting. In addition, computer conferees may fill in questionnaires, take part in Delphi exercises or fill in cross-impact matrices; such aids to the collection of expert opinion are usually too cumbersome for live conferences. 6b6

6b7 When live conferences become very large, it is common for subcommittees to be formed or for the conference to be broken into several subconferences. Again most teleconferencing systems have similar capabilities. 6b7

6b8 Finally, live conferences tend to be episodes in an organization's general communication process. They cannot easily be separated from other conferences, from general information flows, or from material in an organization's files. Moreover, every conference has a pre-meeting stage and an implementation stage, in which characteristic communications take place. 6b8

6b9 So far, teleconferencing systems have been built almost exclusively to handle communication processes for an individual meeting. 6b9

6b10 Overall, state-of-the-art teleconferencing systems handle the bulk of interaction processes desirable in individual meetings. They also add several new dimensions to meetings: the conference record lets people participate when convenient for them; a number of voting-like tools can be made available to conferees; and some systems allow documents and data files to be entered into the record. None of the systems built to date allow conferences to be integrated easily into the general communication flows of an organization. 6b10

6c 2. Sophistication 6c

6c1 When several systems are viewed side-by-side, it is intuitively obvious that some are more sophisticated than others. Sophistication seems to lie along four main dimensions: 6c1

6c2 a. Ease of Use 6c2

6c2a It is generally acknowledged that systems should be easy to learn and use, and all system designers claim that their systems are easy to use. However, systems vary greatly in their approaches to ease of use. The only universal proposition is that the systems should be "bug" free--that it should perform as specified in its documentation. 6c2a

6c2b There appears to be some trade-off between ease of use and power of the system. A system that simply gives the user multiple choices may be simple to learn; provided choices are clear, but such systems may be tedious to use. Systems that allow many user commands may take more time to learn but may be preferred in the long run. In addition, more powerful systems generally provide more on-line help to the user. So even the power/ease-of-use tradeoff is unclear. In addition, there are no methodologies to decide objectively whether a given system is easy to learn, much less to decide whether more power is desirable. 6c2b

6c2c Overall, there appears to be an elusive quality of "elegant design" that characterizes the best systems. This is usually manifest in small but important ways. For example, in FORUM/PLANET, the conferee is given a list of conferences open to him or her. If there has been activity in one since he or she last entered it, an asterisk is printed before its name. 6c2c

6c3 b. Design and Analysis Spaces 6c3

6c3a All systems include a set of commands for conferees. The best systems also give coherent sets of procedures to the conference designer (the chair person) and those who wish to analyze interaction patterns retrospectively 6c3a

6c4 c. Focusing and Reading 6c4

6c4a Conference records may become very long--consisting of several thousand entries. Unless steps are taken to keep the discussion focused on its target, the conference may ramble chaotically. In addition, reading aids to help participants view the record become mandatory. 6c4a

6c4b Focusing is usually accomplished by breaking a conference into subconferences, each with a more limited topic area. Voting and related tools also serve to focus the conference. One technique for focusing has not been used to its fullest extent. This is to base the conference on a position paper or series of papers. Comments could then be addressed to sections of the papers. 6c4b

6c4c Subdividing conferences into subconferences also aids viewing. The number of comments in the subrecord becomes manageable for viewing. 6c4c

6c4d In addition, all systems simplify the viewing task by allowing the viewer to create sublists of entries. For example, the participant may wish to view all entries since he or she last left the conference. Specifications may be rather complex. For example, the participant may wish to view only comments by a certain author, made after a given date, referencing a certain prior entry and containing a given text string. In essence, the participant is creating a transient edited copy of the record, filtered by the specifications. 6c4d

6c4e Filtering can also be active. A participant may leave his or her terminal on but specify which comments will appear. For example, only comments made by a certain author may be printed or displayed or perhaps only private messages directed to the participant. 6c4e

6c4f Finally, filtering can be regularized. Special subrecords may be kept to which comments with given characteristics are appended, whether or not the participant is currently logged in. This is essentially an automatic filing system. 6c4f

6c4g Several other viewing aids have been used in teleconferencing packages. We have discussed filtering primarily to illustrate the degree of elaboration present in state-of-the-art teleconferencing systems. Although no system has all of these filtering procedures, almost all teleconferencing systems have most. 6c4g

6d 3. Limitations 6d

6d1 As a group, teleconferencing systems have three chief limitations that will compromise, to some extent, any experiment to assess their role in organizations. 6d1

6d2 First, there is no agreement on what a teleconferencing system should look like. Designers are still debating many points, and new ideas are still emerging at a rapid rate. Teleconferencing systems of the the future are likely to be considerably more sophisticated than those of today, a factor to keep in mind during evaluations. 6d2

6d3 Second, costs are still high. Each participant hour will cost, generally, between $15 and $20, not including local communications to the network or computer, and not including terminal rental. Costs are likely to decline in the future, but their rate of decline is a matter for conjecture. 6d3

6d4 Third, future teleconferencing systems may be more effective, because they will be combined with a general computer communications system for an organization. Some synergy should emerge if conferees are familiar with the system and can refer to appropriate files and documents easily during conferences. 6d4

7 CASES: 7

7a ORACLE 7a

7b CONFERENCE/RIMS 7b

7c CONFER 7c

7d FORUM/PLANET 7d

7e General Conferencing System 7e

7f CMI (?) 7f

7g Others 7g

7g1 Turoff Offshoots 7g1

7g2 Scrapbook's confer 7g2

8 8

9 QUESTIONNAIRE ON COMPUTER TELECONFERENCING FOR NLS ARCHITECTS 9

9a The purpose of this questionnaire is to ascertain in a brief way, the present interest in and familiarity with the subject of computer teleconferencing among Augmentation Research Center/NLS users. Initial research into the subject graphically illustrates that a number of computer systems now support various versions of teleconferencing software. Our interest in pursuing the topic is to develop a composite picture of the interest of NLS users as a group toward having teleconferencing facilities available at Office-1, either within (i.e. as a subset of) NLS, or as a separate software program on Office-1, -2, etc. 9a

9b The present document and its attachment are the first attempt in the above direction. Hopefully, the feedback will make it possible to convey to ARC developers a definite statement about the desirability, general characteristics, and perceived priority of implementing teleconferencing capabilities within the ARC system. 9b

9b1 Q1. Research to date indicates that there have been at least 8-10 separate attempts around the country to develop and implement computer conferencing packages. These include FORUM, PLANET, CONFER (PLATO IV) PARTY LINE, CONFERENCE/RIMS (Turoff), EMISSARY, ORACLE, TCTALK (Calvin) and General Conferencing System (GCS). (See attachment for background and details). Have you ever used any of these systems or any other teleconferencing system? 9b1

9b2 Q2. If yes, which one or ones? 9b2

9b3 Q3. How much experience have you had with such systems, i.e., number of times used, duration of use, etc.? Approximately when (i.e., calendar dates)? How many others participated? 9b3

9b4 Q4. Are you convinced that teleconferencing should be available to NLS users on Office-1, -2, etc.? If so, could you tell us briefly why, or why not? 9b4

9b5 Q5. What do you think you would use it for? 9b5

9b6 Q6. How would this use be different from mailbox capabilities such as message transfer in TENEX and the NLS journal system that are already implemented? 9b6

9b7 Q7. Based on your experience, if any, with computer conferencing, which system(s) that you know of best represent what you think a teleconferencing system should be. 9b7

9b8 Q8. From the brief review of teleconferencing attached to this document, can you identify features that you think would be especially useful in an ARC teleconferencing package? Any specific features not especially useful? 9b8

9b9 Q9. Considering other possible improvements, embellishments, etc., that ARC development could be spending its time on over the next 12 months, how would you rank the addition of teleconferencing? 9b9

9b9a Urgent
High
Medium
Low 9b9a

9b10 Q10. Would you favor getting an existing teleconferencing package onto Office-1, -2, etc. as soon as possible (if possible), or the development of a package specifically written for KWAC, by ARC research and development programmers and staff? 9b10

9b11 Q11. If the latter (Q10), would your organization be willing and in a position to contribute financially toward the development of an ARC-based (or NLS-based) package? 9b11

9b12 Q12. Do you have any other comments, questions, gentle cajolings or whatever to add to the discussion? 9b12

9b13 Thanks very much! 9b13

(J32773) 18-JUN-75 07:22;;;; Title: Author(s): Inez M. Mattiuz/IMM; Distribution: /GCE([INFO-ONLY]) ; Clerk: IMM; Origin: <GJOURNAL>32360.NLS;1, 22-APR-75 18:41 XXX ;;;; Title: Author(s): Glenn A. Sherwood/GAS2; Distribution: /TELECON([ACTION]) KWAC([INFO-ONLY]) ; Sub-Collections: NIC TELECON KWAC BELL-CANADA; Clerk: GAS2; Origin: < SHERWOOD, CONF.NLS;12, >, 22-APR-75 17:50 GAS2 ;;;;####;

####;

TI Terminals for NSRDC

DLS 18-JUN-75 12:16 32777

Excuse the delay...have been on jury duty this week.

TI Terminals for NSRDC

1 Ref your messaage (JOURNAL, JRNL27, J32743:gw) concerning TI-735 terminals. The original message was sent to JCN and MEH (Jim Norton and Martin Hardy) at SRI. Their answer was not explicitly given, but is YES...they can and will supply TI and other terminals. The contract for Office-1 contains an optional clause, under which SRI agrees to supply terminal equipment and COM services. The intent was to exercise this option as soon as the contract was signed. However, there was a delay in signing the Office-1 contract with SRI, which pushed us too close to the end of the fiscal year to process the paperwork to accomplish this with '75 funds. We should be OK though, for '76 funds. When the terms of the optional clause have been negotiated with SRI (should be within the next 2 weeks), I'll pass the info on to you. 1

TI Terminals for NSRDC

(J32777) 18-JUN-75 12:16;;;; Title: Author(s): Duane L. Stone/DLS;
Distribution: /RDA([INFO-ONLY]) ELF([INFO-ONLY]) ;
Sub-Collections: RADC; Clerk: DLS;

1 ARPA 12-APR-75 thru 7-JUN-75   7.39   503.51   1.47%   1

1a No data available for weeks of April 5 and 26, May 24.   1a

1b ARPA (80C)   7-JUN-75   .92   56.10   1.64%   1b

1b1 ARPA-PM	7-JUN-75	.03	2.63	1.27%	1b1
1b2 BANGERT	7-JUN-75	.28	14.82	1.88%	1b2
1b3 BECKER	7-JUN-75	.00	.02	4.55%	1b3
1b4 BLUE	7-JUN-75	.00	.05	1.63%	1b4
1b5 CARLSON	7-JUN-75	.21	12.45	1.72%	1b5
1b6 COLEMAN	7-JUN-75	.08	2.97	2.72%	1b6
1b7 DCLEMENTS	7-JUN-75	.02	1.85	1.26%	1b7
1b8 DORIS	7-JUN-75	.00	.03	3.39%	1b8
1b9 DSDC-XF	7-JUN-75	.00	.44	.75%	1b9
1b10 DSDC-SG	7-JUN-75	.16	11.79	1.34%	1b10
1b11 DSDC-SC	7-JUN-75	.00	.06	3.43%	1b11
1b12 DSDC-SYO	7-JUN-75	.00	.03	7.00%	1b12
1b13 DUBOIS	7-JUN-75	.00	.35	.32%	1b13
1b14 EDWARDS	7-JUN-75	.01	.30	2.94%	1b14
1b15 GROSS	7-JUN-75	.05	2.71	1.72%	1b15
1b16 IWWSS	7-JUN-75	.00	.07	1.17%	1b16
1b17 KAHN	7-JUN-75	.00	.39	.71%	1b17
1b18 LICKLIDER	7-JUN-75	.00	.02	5.41%	1b18
1b19 LYONS	7-JUN-75	.00	.03	2.46%	1b19
1b20 MACROS	7-JUN-75	.02	.45	4.10%	1b20
1b21 MCLINDON	7-JUN-75	.00	1.28	.37%	1b21
1b22 ORSINI	7-JUN-75	.00	.20	1.82%	1b22

1b23	RUSSELL	7-JUN-75	.05	2.65	1.87%	1b23
1b24	*bad* *bad* VANDERBURGH					1b24
1b25	WALKER	7-JUN-75	.01	.51	1.69%	1b25
1c	ARPA (800)	31-MAY-75	.84	43.93	1.91%	1c
1c1	ARPA-PM	31-MAY-75	.02	.84	2.02%	1c1
1c2	BANGERT	31-MAY-75	.22	11.98	1.83%	1c2
1c3	CARLSON	31-MAY-75	.04	2.98	1.24%	1c3
1c4	COLEMAN	31-MAY-75	.14	4.99	2.78%	1c4
1c5	DSDC-SG	31-MAY-75	.04	5.27	.77%	1c5
1c6	DSDC-SC	31-MAY-75	.00	.28	.40%	1c6
1c7	GROSS	31-MAY-75	.01	.57	2.23%	1c7
1c8	IWWSS	31-MAY-75	.00	.05	5.45%	1c8
1c9	KAHN	31-MAY-75	.02	.74	3.00%	1c9
1c10	LICKLIDER	31-MAY-75	.00	.03	6.50%	1c10
1c11	MACROS	31-MAY-75	.03	.61	4.89%	1c11
1c12	MCLINDON	31-MAY-75	.00	.58	.62%	1c12
1c13	ORSINI	31-MAY-75	.00	.03	5.45%	1c13
1c14	RUSSELL	31-MAY-75	.32	14.90	2.12%	1c14
1c15	*bad* *bad* VANDERBURGH					1c15
1c16	XGP	31-MAY-75	.00	.08	4.90%	1c16
1d	ARPA (800)	17-MAY-75	1.06	64.19	1.65%	1d
1d1	ARPA-PM	17-MAY-75	.08	3.51	2.16%	1d1
1d2	BANGERT	17-MAY-75	.11	7.67	1.38%	1d2
1d3	BLUE	17-MAY-75	.00	.07	3.44%	1d3
1d4	CAMPBELL	17-MAY-75	.00	.02	3.45%	1d4

1d5	CARLSON	17-MAY-75	.13	8.84	1.49%	1d5
1d6	COLEMAN	17-MAY-75	.08	5.88	1.35%	1d6
1d7	DCLEMENTS	17-MAY-75	.00	.01	8.33%	1d7
1d8	DONCHIN	17-MAY-75	.00	.01	11.11%	1d8
1d9	DSDC-XF	17-MAY-75	.16	7.84	2.00%	1d9
1d10	DSDC-SG	17-MAY-75	.00	.14	2.00%	1d10
1d11	DSDC-SC	17-MAY-75	.07	4.14	1.64%	1d11
1d12	DUBOIS	17-MAY-75	.00	.35	.32%	1d12
1d13	HARRIS	17-MAY-75	.04	3.02	1.17%	1d13
1d14	IWWSS	17-MAY-75	.13	4.67	2.68%	1d14
1d15	KAHN	17-MAY-75	.01	.25	2.37%	1d15
1d16	LICKLIDER	17-MAY-75	.00	.04	5.16%	1d16
1d17	LUDWIG	17-MAY-75	.01	.53	1.21%	1d17
1d18	LYONS	17-MAY-75	.00	.41	1.14%	1d18
1d19	MCLINDON	17-MAY-75	.03	2.05	1.60%	1d19
1d20	ORSINI	17-MAY-75	.00	.35	.39%	1d20
1d21	RUSSELL	17-MAY-75	.00	.43	1.10%	1d21
1d22	STO	17-MAY-75	.00	.03	4.90%	1d22
1d23	*bad* *bad* VANDERBURGH					1d23
1d24	WALKER	17-MAY-75	.19	13.60	1.37%	1d24
1d25	WILLIS	17-MAY-75	.00	.02	6.06%	1d25
1d26	XGP	17-MAY-75	.02	.31	5.98%	1d26
1e	ARPA (800)	10-MAY-75	.90	79.11	1.14%	1e
1e1	ARPA-PM	10-MAY-75	.01	.33	3.19%	1e1
1e2	BANGERT	10-MAY-75	.12	7.90	1.52%	1e2

1e3 BEARD	10-MAY-75	.00	.01	10.00%	1e3
1e4 BECKER	10-MAY-75	.00	.02	5.08%	1e4
1e5 CARLSON	10-MAY-75	.22	11.18	1.96%	1e5
1e6 COLEMAN	10-MAY-75	.01	.29	2.02%	1e6
1e7 CROCKER	10-MAY-75	.01	.24	4.17%	1e7
1e8 DCLEMENTS	10-MAY-75	.01	.14	4.66%	1e8
1e9 DONCHIN	10-MAY-75	.00	.08	2.67%	1e9
1e10 DSDC-XF	10-MAY-75	.23	34.47	.68%	1e10
1e11 DSDC-SG	10-MAY-75	.01	1.49	.95%	1e11
1e12 DSDC-SYD	10-MAY-75	.00	.16	2.47%	1e12
1e13 DSDC-SC	10-MAY-75	.04	1.21	3.50%	1e13
1e14 HARRIS	10-MAY-75	.00	.83	.60%	1e14
1e15 KAHN	10-MAY-75	.02	1.27	1.68%	1e15
1e16 LICKLIDER	10-MAY-75	.00	.16	3.14%	1e16
1e17 LUDWIG	10-MAY-75	.01	.87	.73%	1e17
1e18 LYONS	10-MAY-75	.01	1.38	.40%	1e18
1e19 MACROS	10-MAY-75	.00	.09	2.99%	1e19
1e20 MCLINDON	10-MAY-75	.01	.38	2.65%	1e20
1e21 ORSINI	10-MAY-75	.00	.70	.39%	1e21
1e22 *bad* *bad* VANDERBURGH					1e22
1e23 WALKER	10-MAY-75	.18	15.83	1.12%	1e23
1e24 XGP	10-MAY-75	.01	.08	7.91%	1e24
1f ARPA (800)	3-MAY-75	1.36	96.23	1.41%	1f
1f1 AFDAA-XO	3-MAY-75	.06	1.60	3.65%	1f1
1f2 ARPA-PM	3-MAY-75	.04	2.44	1.83%	1f2

1f3	*bad* *bad* ARPA-PRACTICE					1f3
1f4	BANGERT	3-MAY-75	.48	30.01	1.60%	1f4
1f5	CARLSON	3-MAY-75	.06	2.80	2.16%	1f5
1f6	COLEMAN	3-MAY-75	.00	.26	1.93%	1f6
1f7	CROCKER	3-MAY-75	.05	2.67	2.01%	1f7
1f8	DCLEMENTS	3-MAY-75	.01	1.15	.65%	1f8
1f9	DONCHIN	3-MAY-75	.01	.10	5.28%	1f9
1f10	DSDC-XF	3-MAY-75	.29	30.15	.95%	1f10
1f11	DSDC-PR	3-MAY-75	.02	1.88	1.24%	1f11
1f12	DSDC-SC	3-MAY-75	.12	6.53	1.78%	1f12
1f13	GROSS	3-MAY-75	.00	.01	9.09%	1f13
1f14	HARRIS	3-MAY-75	.00	.41	.34%	1f14
1f15	JACKSON	3-MAY-75	.01	.08	6.79%	1f15
1f16	KAHN	3-MAY-75	.01	.34	3.79%	1f16
1f17	LUDWIG	3-MAY-75	.00	.03	4.72%	1f17
1f18	MCLINDON	3-MAY-75	.02	3.16	.52%	1f18
1f19	ORSINI	3-MAY-75	.00	.34	.24%	1f19
1f20	RUSSELL	3-MAY-75	.16	9.47	1.70%	1f20
1f21	*bad* *bad* VANDERBURGH					1f21
1f22	WALKER	3-MAY-75	.02	2.44	.82%	1f22
1f23	WILLIS	3-MAY-75	.00	.01	4.76%	1f23
1f24	YEE	3-MAY-75	.00	.35	.24%	1f24
1g	ARPA (800)	19-APR-75	1.60	109.34	1.46%	1g
1g1	AFDAA-XO	19-APR-75	.00	.25	1.99%	1g1
1g2	ARPA-PM	19-APR-75	.24	16.51	1.45%	1g2

1g3	BANGERT	19-APR-75	.17	11.96	1.43%	1g3
1g4	BECKER	19-APR-75	.00	.26	1.48%	1g4
1g5	CARLSON	19-APR-75	.07	3.61	1.82%	1g5
1g6	COLEMAN	19-APR-75	.02	1.47	1.19%	1g6
1g7	DCLEMENTS	19-APR-75	.01	.56	2.43%	1g7
1g8	DONCHIN	19-APR-75	.00	.03	5.79%	1g8
1g9	DSDC-XF	19-APR-75	.27	33.41	.82%	1g9
1g10	DSDC-SC	19-APR-75	.22	10.57	2.13%	1g10
1g11	EDWARDS	19-APR-75	.13	7.92	1.63%	1g11
1g12	JACKSON	19-APR-75	.00	.29	1.07%	1g12
1g13	KAHN	19-APR-75	.00	.02	12.82%	1g13
1g14	LICKLIDER	19-APR-75	.00	.02	5.08%	1g14
1g15	LUDWIG	19-APR-75	.00	.54	.77%	1g15
1g16	LYONS	19-APR-75	.00	.10	4.88%	1g16
1g17	MACROS	19-APR-75	.39	12.52	3.08%	1g17
1g18	MCLINDON	19-APR-75	.03	4.43	.73%	1g18
1g19	ORSINI	19-APR-75	.00	.22	1.41%	1g19
1g20	ROMNEY	19-APR-75	.00	.01	7.69%	1g20
1g21	RUSSELL	19-APR-75	.04	2.26	1.93%	1g21
1g22	STO	19-APR-75	.00	.05	3.23%	1g22
1g23	STUBBS	19-APR-75	.00	.19	1.46%	1g23
1g24	*bad* *bad* VANDERBURGH					1g24
1g25	WALKER	19-APR-75	.01	2.00	.54%	1g25
1g26	WILLIS	19-APR-75	.00	.01	6.45%	1g26
1g27	XGP	19-APR-75	.00	.13	3.79%	1g27

1g28 YEE	19-APR-75	.00	.00	11.76%	1g28
1h ARPA (800)	12-APR-75	.71	54.61	1.30%	1h
1h1 AFDAA-XO	12-APR-75	.00	.11	1.80%	1h1
1h2 ARPA-PM	12-APR-75	.01	.16	3.94%	1h2
1h3 BANGERT	12-APR-75	.08	7.03	1.19%	1h3
1h4 BLUE	12-APR-75	.00	.05	3.57%	1h4
1h5 CARLSON	12-APR-75	.04	3.07	1.39%	1h5
1h6 DCLEMENTS	12-APR-75	.01	.07	8.43%	1h6
1h7 DSDC-XF	12-APR-75	.26	26.24	1.00%	1h7
1h8 DSDC-SC	12-APR-75	.12	7.11	1.69%	1h8
1h9 DSDC-SYO	12-APR-75	.01	.23	3.03%	1h9
1h10 DUBOIS	12-APR-75	.00	.34	.32%	1h10
1h11 GROSS	12-APR-75	.00	.01	6.45%	1h11
1h12 HARRIS	12-APR-75	.00	.03	3.42%	1h12
1h13 IWWSS	12-APR-75	.01	.22	3.43%	1h13
1h14 JACKSON	12-APR-75	.00	.26	.86%	1h14
1h15 KAHN	12-APR-75	.00	.16	2.86%	1h15
1h16 LICKLIDER	12-APR-75	.00	.06	3.60%	1h16
1h17 LUDWIG	12-APR-75	.01	.27	3.16%	1h17
1h18 LYONS	12-APR-75	.00	.01	8.00%	1h18
1h19 MACROS	12-APR-75	.06	1.82	3.27%	1h19
1h20 MCLINDON	12-APR-75	.02	.90	1.74%	1h20
1h21 RUSSELL	12-APR-75	.05	2.67	1.81%	1h21
1h22 STO	12-APR-75	.00	.12	1.63%	1h22
1h23 STUBBS	12-APR-75	.00	.17	2.01%	1h23

1h24	*bad* *bad* VANDERBURGH					1h24
1h25	WALKER	12-APR-75	.03	3.37	.92%	1h25
1h26	YEE	12-APR-75	.00	.13	1.25%	1h26

ARPA Usage of Office-1: Preliminary

(J32779) 18-JUN-75 14:10;;;; Title: Author(s): Raymond R. Panko, James C. Norton/RA3Y JCN; Distribution: /CKM([INFO-ONLY]) ; Sub-Collections: SRI-ARC NIC; Clerk: RA3Y;

DAP 18-JUN-75 14:43 32780

The Whole Universe Catalog: a new tool

len, this is a real goodie I received today, thought you'd appreciate it.

DAP 18-JUN-75 14:43 32780

The Whole Universe Catalog: a new tool

1 What it is 1

1a The Whole Universe Catalog (WUC [rhymes with luke]) is a way of looking at things. It consists of a simple accessing system with which you can view from a display or typewriter terminal your personal NLS files as well as the Whole Universe Catalog index. The index currently interfaces to many things of interest to NLS users. If the thing you want is online and you know it's name, WUC can take you to it. If you don't know it's name, you should be able to find it through a synonym or by pointing to categories if it is cataloged in WUC. WUC is meant to eventually become an online university with access to the universe. Hence the ambitious name. 1a

1b The accessing system requires no knowledge of viewspecs, addresses, commandwords, directories or file boundaries. Yet for those who know them, WUC contains all of the addressing and viewing capablities of NLS. It can reduce from ten to one the number of buttons to be pushed for some of the most common viewing functions. From a display terminal it can be used without ever touching the keyboard or keyset. However, any legal NLS ADDRESS can be typed. In DNLS, prompting represents buttons and combinations of buttons on the mouse. Type questionmark in WUC for a short description of the major alternatives. 1b

1c Since WUC has not been accepted as an official ARC program, you cannot type <CTRL-Q> and get help with it or learn about it in help. Instead, type HELP in the wuc command. Also, please send any comments to ident KIRK or sndmsg to KELLEY. Do not send any feedback concerning WUC to FEEDBACK. 1c

2 How to get it 2

2a Use the "Process Branch" command on the following branch (2B) to make WUC available to you. 2a

2b Execute Programs Load Program XPROGRAMS,WUC 2b

2b1 Execute Use IPXPROGRAMS,WUCEW 2b1

2c In DNLS, push the two rightmost mouse buttons together to go NEXT from where you point and thereby scroll through the rest of this document. 2c

2d CD (CTRL-X) quits WUC. Type ew to "Execute WUC" once you have quit. 2d

3 Features 3

3a With WUC you use the same command for the same viewing function in DNLS as in TNLS. 3a

3b WUC allows viewspecs when you want to input them (as in Jump to Link) but does not prompt for them every time. Only one confirm is necessary because it also specifies your view. The confirm can be thought of as a very special viewspec which also acts to terminate the command. 3b

3c Any NLS address is acceptable but if you wish to address a name in a certain branch, you need not preceed it with exclamation point. 3c

3d If you type a long address and misspell the last word, WUC will take you as far as it can and indicate the rest of the address was not found thus saving you from re-typeing the beginning of the address. 3d

3e Parenthetical comments are not confused with links. Text in parentheses is ignored unless you use the .l address element. 3e

3f Locating Documents 3f

3f1 WUC makes document locator files and super documents easier to use in several ways. Probably the most important to the novice is that you do the same simple things to 1) get to the locator, 2) search down through it, 3) access a particular file and 4) find the piece of information you want in that file. In addition, WUC eliminates five of the rather criptic buttons necessary most of the time in TNLS (pbSN.l<CA><CA>). With WUC in TNLS, you just type the statement number (SN) and confirm. In DNLS this is a single mouse button push. 3f1

3f2 For instance, to access the userguides locator, simply type LOCATOR followed by your CA. If you type LOCATOR OUTPUT you go directly to the output processor guide skipping any view of the locator all together. If you type LOCATOR OUTPUT PES you go directly to the description for the PES output processor directive. In this case all you have to type is OP PES because the output processor guide happens to be available directly from the WUC index under the name OP. 3f2

3g Advantages of WUC over the Help command. 3g

3g1 For help with NLS use WUC and type NLS. You can also go directly to a particular definition or branch of information by typing NLS followed by the name of the information. For example, NLS LINK. 3g1

3g2 WUC contains help for all user-programs made available in WUC by their authors instead of just the ones approved by ARC management. Type PROGRAMS. 3g2

3g3 WUC allows you to scroll to read everything about a certain topic at once without having to back up and try each menu one at a time. When you do want to back up, it is just one push of a button (BC) and in TNLS, only enough is typed to indicate where you have returned. Where you go when you back up is not confusing. Backing up ALWAYS takes you to the next place back. It remembers your last 60 places on a simple stack. 3g3

3g4 You can see everything or just an outline at your discretion. 3g4

3g5 In DNLS, you use the mouse to point to a menu item or point to a word for its definition instead of having to type it in from the keyboard. 3g5

3g6 WUC accepts any valid NLS link, address, viewspec, or content analyzer pattern or program in addition to the simple words accepted by Help. You never get the cryptic message "Illegal list characters" while using WUC. 3g6

3g7 WUC will take you through as many links and files as necessary to get you to where the Help description writer wishes to take you. Even if you type "science fiction" and the information is located under "novels" which is linked to from "fiction" located under "sciences" linked to from "science", WUC will find it. The Help command will not search through links. Also, if the Help database links to a link, you see nothing when you use the Help command. 3g7

3g8 Help Description Writers: you can stop right where you are and edit your current view. 3g8

3g9 Since WUC can access any NLS file with the essential flexibility needed, the current restrictions on format are unnecessary. The ugly, chopped up help database we now have, could be a smoothly flowing online document which would map almost directly into a discursive prose hard copy document. 3g9

3g10 WUC allows you to have your current view printed on the line printer. 3g10

3h Query 3h

3h1 Although it is currently not available for Novice users, it would be very easy to access, for example, the Arpanet Resource

Notebook using WUC. The current "Bring" and "Show" commands as well as the instructions for use duplicated in every file would be unnecessary. Typing questionmark should be sufficient to teach the user what to do. 3h1

3h2 The user could type, for example, NIC SRI-ARC PERSONNEL and see the information directly without having to wade through the current laborious path. 3h2

3h3 All of the writing advantages listed in the Help discussion above are as important for Query database builders. Standard online documents could be written for the database and turned directly into hard copy with very little effort. 3h3

3i If you try to take an invalid link, WUC will indicate the problem and display the statement containing the invalid link. 3i

3j You can go "up" to a higher level from the origin of a file if a link to the source for that file has been specified in the origin. This helps make many files accessible as a single "virtual" file. Such a file is merely a special branch in a theoretically unlimited file and is treated as any other branch. 3j

3k You generally get faster response with WUC because the computer doesn't have to handle as much user interaction before it calls core routines. 3k

3l In WUC, type WUC for a list of the things available in the WUC index. 3l

3m Bon Voyage, information space person. 3m

The Whole Universe Catalog: a new tool

(J32780) 18-JUN-75 14:43;;;; Title: Author(s): David A. Potter/DAP; Distribution: /LCS([INFO-ONLY]) ; Clerk: DAP; Origin: < HJOURNAL, 32768.NLS;1, >, 18-JUN-75 10:43 XXX ;;;; Title: Author(s): Kirk E. Kelley/KIRK; Distribution: /LHD([INFO-ONLY]) GCE([INFO-ONLY] DVN said your might be receptive to this) DAP([INFO-ONLY]) GAS2([INFO-ONLY]) PWO([INFO-ONLY]) DLS([INFO-ONLY]) NJN([INFO-ONLY]) DHC([INFO-ONLY]) WKE([INFO-ONLY]) LPD([INFO-ONLY]) JTM([INFO-ONLY]) DCW([INFO-ONLY]) WSD([INFO-ONLY]) RWW([INFO-ONLY]) JAC3([INFO-ONLY]) DMB([INFO-ONLY]) LLG([INFO-ONLY]) BEV([INFO-ONLY]) ARC-APP([INFO-ONLY]) ; Sub-Collections: SRI-ARC ARC-APP NIC; Clerk: KIRK;

1974 Dialog on RADC JOVIAL Manual

1 (RADC) Jovial documentation 1

1a 3-OCT-74 1957-PDT VANNOUHUYS: JOVIL Manual
Distribution: STONE AT OFFICE-1, meyer, norton
Received at: 3-OCT-74 19:57:59 1a

1a1 The last time I heard from you was your question about the wehere abouts of the copyflow of the file EQU. I replied that it hd been mailed to you the day before. I asume that the reason I have no heard from you sine is that you got the proofs and are considering them. 1a1

1b 17-AUG-74 1707-PDT VANNOUHUYS: Another Jovial Draft
Distribution: STONE AT OFFICE-1, norton, meyer
Received at: 17-AUG-74 17:07:54 1b

1b1 I just made (stone,comtest,) at office one into a com file , AUG17JOVTRY, and moved it to ISI and then oonto tape for DDSI to pick up. I noted that the file was 117 pages in NLS (81 as a sequentila files). I assume that when you spoke of "400" page syou ment via xcom. 1b1

1c 16-AUG-74 1255-PDT STONE at OFFICE-1: Jovial manual test run...second try
Distribution: VANNOUHUYS, vannouhuys at sri-arc, meyer at sri-arc
Received at: 16-AUG-74 12:55:30 1c

1c1 I have the secnd run ready..as ready as I can make it. Its in a file called <stone>comtest. Please send it to USC-ISI for me...I have not master that part of the process yet. I've made an XCOM run several times and it looks good to me. If you want you can make one too, but its 400+ pages. Good luck, I will be on travel the next week and arrive (according to current plan) at SRI on the 26th (along with Jo Cavano). See you then. If you get printouts from DDSI, keep them here and we can go ver them together.
Thanks
Stoney 1c1

1d 15-AUG-74 1125-P DVN: SRI Proposal ISU 74-165 - Publication Support for the Jovial J73 Programming Language Specification Document
Distribution: DVN DLS JCN EKM NDM RWW
Received at: 15-AUG-74 12:59
Location: (MJOURNAL, 23801, 1:w) 1d

1e 6-AUG-74 0744-P NDM: Answer to 30944: Output Processor Directive Problem

Distribution: NDM DLS EJK
Received at: 6-AUG-74 08:32
Journal Message 30973 1e

1e1 Duane and Ed: Sorry to be so long in getting back to you on your directives problem. You were right to look at the PxF directives, but you didn't quite get them right. Those guys are pretty tricky. Try:

The Output Processor Users' Guide lists the defaults for those three parameters, and shows their general form. Let me know if you have any further trouble with the little devils! --Dean 1e1

1f 24-JUL-74 1037-P EJK: Output Processor Directive Problem
Received at: 24-JUL-74 15:51
Location: (GJOURNAL, 30944, 1:w) 1f

1g 5-AUG-74 1250-PDT MEYER: JFORM3 and Content Analyzer Programs
Distribution: DECONDE, stone, meyer
Received at: 5-AUG-74 12:50:05 1g

1g1 When you use a content analyzer program which edits the file it works on, it usually won't display/print anything. You were correct to use Goto Programs Get JFORM3<CA>. You'll notice it will be automatically instituted as the current content analyzer filter. 1g1

1g2 In DNLS, if you recreate display (by a Jump or with viewspec-f) with viewspec i on, it will begin to work on the statement at the top of the screen. Unless you use viewspecs g or l, it will do every statement in the file. In TNLS, you control which statements it works on by sazy Print ZBranch or Print Plex... 1g2

1g3 While it is working, JFORM3 will display/print "Catalog File" for each citation it reformats. I know that's not very pretty but so it goes. In DNLS, when it is done, you will be left with an "Empty" on the screen. In TNLS, you'll get the herald. 1g3

1g4 In TNLS, you could then print the branch with viewspec-j. In DNLS, enter viewspecs "hjf" (type those characters while holding the left two buttons down on the mouse). 1g4

1g5 Goodluck. Let me know if you run into any further problems. --Dean 1g5

1h 29-JUL-74 1833-PDT FEEDBACK: sendprint
Distribution: CARRIER, meyer, feedback
Received at: 29-JUL-74 18:33:39 1h

1h1 Bobbie, We are working on the problem...it is a hard one.

Need to know if this is your first use of the feature or if the stop at page end has worked before and this is a new bug. Jim 1h1

1i 23-JUL-74 0820-PDT CARRIER: SENDPRINT
Distribution: FEEDBACK, carrier, meyer, bair
Received at: 23-JUL-74 08:20:19 1i

1i1 We are asking SENDPRINTto wait at page breaks. It does so, but the leading spaces in the first line of the new page disappear. Tried on two different terminals. Sendprint is eating them. --Dean and Bobbie Carrier (RADC) 1i1

1j 16-JUL-74 1542-P DVN NDM EKM DLS: COM Problems: JOVIAL and NSW Manual and DDSI'S Faileur to make 48x Fiche
Distribution: DVN NDM EKM DLS DPCS
Received at: 16-JUL-74 16:30
Location: (GJOURNAL, 23632, 1:w) 1j

1j1 Comments: This is a collection of sendmssages journalized for the record. 1j1

1k 26-JUN-74 1042-P DVN: State of COM: JOVIAL Manual, New Character Widths, Microfiche of NSW Manuals.
Distribution: DVN NDM DLS DCE RWW DDSI DPCS COM
Received at: 26-JUN-74 12:31
Location: (GJOURNAL, 23477, 1:w) 1k

1l 14-JUN-74 1200-PDT VANNOUHUYS: JOVIAL PROGRESS
Distribution: STONE AT OFFICE-1, meyer, norton
Received at: 14-JUN-74 12:00:58 1l

1l1 Glad to hear the samples look good. We await your next trial chapter.
On the tabstops, Dean is here for a week. I have asked him to call you. 1l1

1m 21-MAY-74 1138-P DVN: Problems wit the JOVIAL Manual, Jammed and Missing Characters: The Ball is back to DDSI.
Received at: 21-MAY-74 14:45
Location: (MJOURNAL, 23070, 1:w) 1m

1n 20-MAY-74 0731-PDT MEYER: JOVIAL test run 2
Distribution: MEYER, stone at OFFICE-1
Received at: 20-MAY-74 07:32:00 1n

1n1 Duane: Yes, level indentation is taken from the left margin. (All but the margin settings themselves are taken from left margin; margin from LMBase.) There is a directive which will turn off names in OP without leaving it to viewspec. Will ask you about state of comtest2 when DDSI ready to do things right. --Dean 1n1

1o 17-MAY-74 1114-P NDM: RADC Inter-Office Memo Format
Distribution: NDM DLS FEED
Received at: 17-MAY-74 11:28
Journal Message 23030 1o

1o1 Duane: No way to do that simply with directives. SRI's standard format is similar, so I wrote a user content analyzer program which does approximately that. It's called SRIFORM. We may be able to modify the format slightly to get just what you want. Take a look at eh program and let us know if you guys need help modifying it. (Note: it does a special thing for statements with bullets, i.e. starting with . or -) You can update a file, reformat it with the program, do an output device printer, the unlock the file. --Dean 1o1

1p 17-MAY-74 0542-PDT STONE(DLS) at OFFICE-1: JOVIAL COM RUN
Distribution: meyerSRI-ARC, michaelSRI-ARC
Received at: 17-MAY-74 14:47:33 1p

1p1 vannouhuys@SRI-ARC 1p1

1p2 The reference to the initial JOVIAL COM test run is in: 1p2

1p3 DLS 24-APR-74 12:45 30546
First Experimental Comm Run
Location: (LJOURNAL, 30546, 1:w) 1p3

1p4 Unless you wish to run this to compare with the results of the first run, I wouldn't bother, since it has served its purpose...giving us a chance to see the effects of font changes etc. I have a file called comtest2 in my directory, which is about ready. For some reason, the Tabstops directive prints. Also, there appears to be too much indentation of the left margins. (the PXI indentation appears to be adding to the LM indentation). I would appreciate your looking at this file and sending it with any fixes that you see fit. If you do send it, turn the statement names Off. 1p4

1q 15-MAY-74 1110-PDT STONE at OFFICE-1: formatting problem
Distribution: MEYER
Received at: 15-MAY-74 11:12:06 1q

1q1 Not related to JOVIAL...interoffice memo thing...We want to not indent the first level, then cummulatively indent the first line of 2-n levlels, but have the 2nd, 3rd, etc lines of the lower levels unindented, ie, lined up with level 1. I've tried several combinations of directives, but so far no luck...any suggestions?? 1q1

1r 15-MAY-74 1104-PDT STONE at OFFICE-1: rerun of JOVIAL test
Distribution: MEYER
Received at: 15-MAY-74 11:05:35 1r

1r1 Unless you want toa make a second run of the test file to verify for your own purposes that the fixes have been made in the News Gothic font, I wouldn't bother. I'd rather send another file, now that some of my directives problems have been cleared up...it will include the equations file, since we can't really start production runs until I have them down pat. Did they give you any time frame on the font fix?? 1r1

1s 30-APR-74 1150-PDT VANNOUHUYS at SRI-ARC: COM TEST
Distribution: MEYER
Received at: 30-APR-74 11:50:38 1s

1s1 The clowns at DDSI had the address of SRI Washington Wrong. 1s1
1t 30-APR-74 1046-PDT VANNOUHUYS at SRI-ARC: First Jovila Run
Distribution: MEYER, stone at NIC
Received at: 30-APR-74 10:46:19 1t
1t1 I got duane's comments, I have sent a copy of the ODP version of the the journalized version to Dean, this morning so he ca foollow them better. Some of the problems are at DDSI and we will have to go back and ask them to fix things. We are noot ready to do that untilwe study the MS more carefully. I will be on vacation from Thursdzy through the 10th, so it will probably be Dean wh o takes the problems to DDSI. It happens I will be in Los Angeles and could go over there if it was useful. 1t1
1u 29-APR-74 1212-P DLS: Problems With Initial JOVIAL Manual COM Run
Distribution: DLS DVN NDM RN2
Received at: 29-APR-74 14:38
Location: (LJOURNAL, 30579, 1:w) 1u
1v 23-APR-74 1432-P DVN: Draft of JOVIAL Manual Arrives at ARC
Distribution: DVN &DPCS DLS NDM
Received at: 23-APR-74 17:12
Journal Message 22806 1v
1v1 It arrived by Parcel Post with the film inclosed. There are some problems with spacing when we change faces. Ther may be some other problems that non-speakers of JOVIAL can't recognize. Let me know if your drafts don't arrive soon. 1v1
1w 18-APR-74 0938-PDT MEYER(NDM) at SRI-ARC: JOVIAL: DefSyn
Distribution: MEYER, VANNOUHUYS, stoneOFFICE-1
Received at: 18-APR-74 09:38:08 1w
1w1 Duane: I looked over the JOVIAL thing. DefSyn can only define synonyms for directives, not for the parameters mnemonics. Other than that, it looks like the COM run now in progress should do approximately what we expect (never can be too sure, and the way to find out is to run a section as we are doing). We'll see! --Dean 1w1
1x 16-APR-74 2214-PDT SRI-ARC at USC-ISI: JOVIAL CHAPTER
Distribution: MEYER, STONE at NIC, BAIR at NIC
Received at: 17-APR-74 08:57:09 1x
1x1 I JUST NOW SPUN THE TAPE OFF AT ISI. DDSI SHOULD PICK IT UP IN THE MORNING
...DIRK 1x1
1y 16-APR-74 1236-PDT VANNOUHUYS: First JOVIAL COM Run
Distribution: MEYER, stone at NIC, bair, norton, vannouhuys
Received at: 16-APR-74 12:36:47 1y
1y1 Duane,
As of this morning Dean had not gotten the hard copy of the (carrier,comtest,) ODP Printout. Unless I hear from you othrwise I am going to make a COM file tonight and sip it to ISI, thence it will go to DDSI in the morning. I just tried to

phone you but could not gett through---the net isthe only
reliable medium of communication left. I will ask DDSI to send
proofs to me here, Dean in Washinton, and to yo at Rome.
Just this minute I coppied (carrier,comtest,) at office-1 to
(documentation,comtest-rem) at ARC and that latter file is the
one I plan to process. 1y1

1z 15-APR-74 0812-PDT VANNOUHUYS: Review of ODP Version of
JOVIAL
Distribution: MEYER, stone at NIC
Received at: 15-APR-74 08:12:17 1z

1z1 Dean,
I sent you the ODP printout of the JOVIAL Chapter Tues or
Wednesday. If you don't get it by today let me know. Maybe in
that case I will go ahead with the COM run, I will talk to
Duane about that. 1z1

1a@ 11-APR-74 1449-P DVN: Plans for JOVIAL Manual
Distribution: DVN DLS NDM JHB &DPCS
Received at: 11-APR-74 17:08
Journal Message 30457 1a@

1a@1 I got Duane's message of April 10th. I will hold further
coment on the state of the directives in (carrier,comtest,)
untill we hear from Dean. I will plan to bill through us. The
second manual sounds as if it's in the fat part of the NLS
strike zone. 1a@1

1aa 10-APR-74 0903-PDT STONE at OFFICE-1: JOVIAL Comtest
Distribution: MEYER, vannouhuys at SRI-ARC, vannouhuys, meyer
Received at: 10-APR-74 09:03:55 1aa

1aa1 I cleaned up <carrier>comtest a little more. Grab
directives are OK now, ... don't worry about the page
numbering...one problem that seems to remain is the ppprinting
of three define directives..I'm not sure why..maybe I have the
wrong format also S for Slant and B for BoldFace.....is
this incorrect? 1aa1

1aa2 I was talking with JCN yesterday, and he says that its OK
to add a task to the existing contract (ARPA/ARC)..not the
utility..to pay for the printing at DDSI. It would take us six
months here to write a new contract..sole source and
everything. One of Nelsons guys is proofing the text now, so
we should be ready to go into production as soon as this test
run is complete..a couple of chapters have been proofed aready,
all the text isnow entered. 1aa2

1ab 9-APR-74 1115-P DVN: Proplems and progres with JOVIAL
Manual
Distribution: DVN DLS NDM
Received at: 9-APR-74 16:35
Location: (LJOURNAL, 30441, 1:w) 1ab

1ac 5-APR-74 1624-P DVN: COMTEST of JOVIAL Chapter
Distribution: DVN DLS RJC NDM JHB DDSI

Received at: 5-APR-74 17:21
Location: (LJOURNAL, 22634, 1:w) 1ac

1ad 5-APR-74 1411-PDT CARRIER at OFFICE-1: COMTEST for JOVIAL Manual
Distribution: MEYER, meyer, vannouhuys, vannouhuys at SRI-ARC, norton
Received at: 5-APR-74 14:12:53 1ad

1ad1 I have a file....contained in <carrier>comtest with directives for com in it. I was going to try to make it a thing o beauty, but time is running out, and since it will be used primarily as a demonstration to Nelson of diferent fonts and sizes, I haven't bothered with page breaks etc....let them fall where ttthey may for the time being.
In doing XCOM, use viewspec D, otherwise use default viewspecs.
I have included a couple of copies of the equations, since the question of copppying them into the body of the text hasn't been resolved in my mind yet. I want to see how bad/easy it is to get them formated correctly. I will leave it up to you guys to send the file to DDSI, unless you think it needs major modifications.
There are only a few pages of Glossary that remain to be inputed...all 11 chapters and the equations have been input and edited by bobbie, me and the newword program. They are currently being edited by one of Nelsons guys, after which any last minute changes will be made and then comes the production job. 1ad1

1ad2 There is a second subsetting manual, which contains descriptions of three subsets of JOVIAL. Its 707 pages long, but appears that there is about a 90% overlap in text and equations. Some judicious copying should allow us to produce that one in "short order".
Good luck....let me know how you make out. 1ad2

1ae 18-MAR-74 1501-P DVN: JOVIAL PROGRESS
Distribution: DVN NDM &DPCS DLS
Received at: 18-MAR-74 16:04
Journal Message 22413 1ae

1ae1 Duane has got two chapters of the JOVIAL Manual journalized as (jjournal,30219,) and (jjournal,30205,). I have printed and ODP and a COMtest version of the former and will mail you copies. 1ae1

1af 5-MAR-74 1826-P NDM: JOVIAL documentation: Production Details
Distribution: NDM DLS DVN JCN
Location: (HJOURNAL, 22232, 1:w) 1af

1ag 28-FEB-74 1700-P DVN: Publishing a JOVIAL Manual through COM, Comments and Questions
Distribution: DVN DPCS COM DLS PR TLH DDSI

Received at: 1-MAR-74 09:22
Location: (HJOURNAL, 22140, 1:w) 1ag
1ag1 Comments: responds to 30111 1ag1
1ah 25-FEB-74 0836-P NDM: JOVIAL Documentation on COM
Distribution: NDM DLS JCN DVN JHB
Journal Message 22043 1ah
1ah1 Duane: The JOVIAL documentation does sound like an
interesting and appropriate application of COM. For the
moment, I am pretty busy with the DEIS stuff, but I would like
to work with you on developing a format when I have time.
Columns necessitate a bit of extra hand formatting, but that
may be OK. It would be very difficult to have an insert
interrupting a columnated page, but it would not be harc if the
text progressed from upper left to upper right, then the insert
then lower left and lower right, if you know what I mean. Once
you and Norton work out the business arrangements, I think the
steps would be 1) talk about what you'd like to see, 2) send
off a few sample formats (they would be printed on microfilm,
then Xerox CopyFlo to get a proof), 3)refine the formats,
4)choose one and order the photo-ready masters (photographic
prints of the microfilm for photo-offset), 5) paste up the
graphics and special characters, and 6) send to a photo-offset
firm (we use in-house SRI printers). We can get together on
all this when you are ready to go. --Dean 1ah1
1ai 21-FEB-74 1133-PDT VANNOUHUYS: Printing Jovial Document
cc: stone at OFFICE-1, norton
Received 21-FEB-74 11:33:46 1ai
1ai1 I've Up to my neck or higher in energy stuff. Just now
read your journal item. It strikes me as a very interesting
undertaking. I will be able to answere many, but not all, or
yor questions. I will try to get answers off today. 1ai1
1aj 19-FEB-74 1439-P DLS: Potential use of COM Output by RADC
Distribution: DLS NDM JHB DVN JCN FEED JLM EJK FJT RFI
Received at: 1-MAR-74 09:23
Location: (HJOURNAL,30111,1:w) 1aj
1aj1 Comments: Would appreciate any guidance, help etc. that
can be given..Thanks.
Secondary Distribution Copy 1aj1

(J32781) 18-JUN-75 15:34;;;; Title: Author(s): N. Dean Meyer/NDM;
Sub-Collections: SRI-ARC; Clerk: NDM;

1 THE WHOLE UNIVERSE CATALOG
An On-Line University with Access to the Universe 1

2 A WAY OF LOOKING AT THINGS
The Whole Universe Catalog (WUC [rhymes with luke]) is a way of looking at things. Imagine an electric typewriter in front of a portable TV on a table in your room. The TV is hooked to a two-way cable which connects to a network of computers. To the side of the typewriter is a hand-sized box with three buttons on top called a "mouse". The mouse has wheels in it which make it easy to roll across the table. As you roll the mouse around with your hand, a small spot on the TV moves around among the following words: 2

2a
Emergency
Health Care
Calendar
Mail
People
News
Want Ads
Yellow Pages
Bank Account
Legal Help
University
Entertainment 2a

2b Moving the spot to "University" and pushing the first of the three buttons on the mouse causes the previous view to be replaced by the following major branches of the University knowledge tree: 2b

2b1 University 2b1

2b1a Assumptions
Logic
Mathematics
Physics
Chemistry
Biology
Psychology
Sociology
Whole Systems 2b1a

2c You can continue pointing and pushing the first mouse button to see "outline" views of any branch or sub-branches. 2c

2d When you point to a word and push the second button on the mouse, the definition of that word appears. This is followed by a full view of the entire branch of information defined by that word. Pushing the first two mouse buttons at the same time displays the next "page" in the subject you have thus specified. 2d

2e Pushing the third button on the mouse takes you back to any view you have previously seen. 2e

3 LINKS TO TOOLS

References or "links" to related information in other subjects appear in the paragraphs. Pointing to these and pushing the first button displays the referenced information. The term "information" here is used loosely to refer to tools such as apples and scissors as well as information services such as games, computer tools, and voice-video links to people. 3

3a These tools include commands which allow you to compose, copy, digest, and produce the personal "information space" in which you can do knowledge work such as generating and publishing your own sections for the Whole Universe Catalog in your field of study. Among some of the tools currently available are one to send mail to those who are on-line and another to have pages automatically printed on the electric typewriter for off-line distribution. 3a

3b To use a tool available via the Whole Universe Catalog, point to or type the name of the tool you want. You are then passed on to that tool until you are finished using it. 3b

4 TYPING WORDS

Typing any word terminated by the Carriage Return key on the electric typewriter causes the definition of the word to appear the same as pointing to a word and pushing the second mouse button. Terminating the word with the first mouse button instead of a Carriage Return causes an "outline" of the branch of information defined by that word to appear. A range of search specifications can be typed in addition to single words. 4

4a Using the typewriter is not necessary to get information. You can access all of the information available via the Whole Universe Catalog simply by pointing to what you want to see with the mouse. A baby on a tricycle with three big buttons on the handle bars could use the Whole Universe Catalog accessing system with appropriately colorful picture information projected on a wall. On the other hand, WUC can be used exclusively from the on-line typewriter if you don't have a mouse. 4a

5 THE UNIVERSITY

The University section of the Whole Universe Catalog attempts to coherently provide a concise, accurate model of the universe with access to alternative models; a place to stand for observing and participating in the process of the universe and improving the range and accuracy of the model; a way of looking at things. 5

5a The University is structured for both a classical and an inter-disciplinary education depending on how the student wishes to learn a subject. Each branch of the knowledge tree points via links to supporting information classified in other branches. This allows the student to ask the inexhaustible "why?". For example, a student with a question about the basis of economics could easily end up taking a coherent path through the branches of politics, psychological motivation, biological psychology, biochemistry, atomic physics, calculus, set theory, and logic before satisfying his curiosity. 5a

5b Our current goal is to provide an interface to the universe which consists of information up to the level of first-year college survey courses. This is expected to take two person-years. 5b

6 THE SPACE-TIME CONTINUUM

One section of the Whole Universe Catalog called the "Space-Time Continuum" allows specification of any space-time location in the universe via pointing to maps in the center of the TV screen, or a list of dates on the side of the screen. At any point, a cultural over view of that place and time is available with the second mouse button along with links to sets of specific things such as music, architecture, news and literature. Within this coordinate system, there is a place for any specific thing and any general concept with many different alternative classification hierarchies for finding information easily. As of this writing, it is mostly a bare outline. It would take a person-year to turn the Space-Time Continuum into a meaningful educational tool. 6

7 VIDEO TAPE OF THE WHOLE UNIVERSE CATALOG.

A 30 minute video-tape of an on-line, real-time journey through the prototype Whole Universe Catalog is available. It takes you through the default cable television user's initial branch including glimpses of the places for news off the AP hot wire, entertainment, medical information, legal aid, shopping, bank accounting, want ads, personal, regicnal, and universal directories of people. A tour of the University follows with a free access education session crossing the inter-disciplinary boundaries of communications, economics, psychology, physiology, biology, chemistry, and nuclear physics. 7

7a There is a session in the Space-Time Continuum. Starting at the origin of the universe you watch it unfold to the present. Zooming down through maps of galaxies, stars, the solar system, to the earth, watch the continents drift as you travel across time to the middle ages. After viewing a textual overview, you move on to the present and view some of the current headlines. Going back to pictorial maps, you zoom in on continents, cities, streets, buildings, into the room where the WUC demonstration is being given, into the protected personal information of the demonstrator including medical records and references to the university that go through organs, into cells, down to molecules and finally to a view of the Hydrogen atom. Towards the end of the tape is a trip to the future plans for the Whole Universe Catalog as contrasted to some predictions by George Orwell. 7a

7b For a copy of the tape, send $30 or a blank 1/2 inch reel of video tape to Kirk Kelley, J2029 SRI, 333 Ravenswood, Menlo Park, CA 94025. 7b

8 GETTING ACCESS

The Whole Universe catalog currently runs on various PDP-10 computers on the ARPA network. Officially the ARPANET is only to be used by DoD contractors though it is possible to find friendly people who will let you use their terminals, passwords, and accounts. We will try to publish a list of these people in the future. 8

8a The Augmentation Research Center sells the services of NLS (the environment in which WUC lives) to people that fit into the Augmented Knowledge Workshop Community. They are especially looking for non-DoD customers who will buy some portion (preferably all) of a "slot" on the computer. These slots currently cost approximately $40,000/year each. (That could be less than 20 cents a minute if it was sold by the minute.) This price is continually falling. Besides computer time, the price includes training in how to use NLS and a say in how it develops. Contact 8a

8a1 The Assistant Director in charge of Applications
Augmentation Research Center
333 Ravenswood
Menlo Park, CA 94025 (415) 326-9716 8a1

8b HARDWARE YOU NEED

You should have a teletype-like computer terminal and/or a display work station. A display work station consists of a video terminal, a mouse, a modem (interface for the terminal to the computer) and a box called a "Line-Processor" which ties the modem, CRT, and mouse together. There are many manufacturers of teletype-like terminals of which Texas Instruments (TI) in San Antonio, Texas is one of the best. Cheap video terminals that work are the Hazeltine "2000", Datamedia "Elite 2500", and Lear Siegler "ADM". They cost about $1000 each or rent for around $100/mo. The mouse ($300) and Line Processor ($2500) are both available from Cybernex, 922 Industrial Av., Palo Alto, California. Datamedia corporation is at 7300 N. Crescent Blvd., Pennsauken, NJ 08110. 8b

9 WEC, LWEC, WUC and WEE

The name "Whole Universe Catalog" was picked to answer the question "What comes after the Last Whole Earth Catalog?" This was some time before the Whole Earth Epilog appeared. 9

9a In a warm-up version of WEE in Harper's wraparound under the title "EDITOR BREAKS PROMISE" Stewart Brand lists four reasons for resurrecting the Catalog. The last one is "After burning our bridges we reported before the Throne to announce, 'We're here for our next terrific idea.' The Throne said, 'That Was It.'" 9a

9b Time to look around for a better throne? 9b

9c It should be noted that WUC has no official relations with WEC or WEE and was conceived and developed to it's current stage independent of the Point foundation (though Richard Austin, a member of Point, did come out and take a look in 1974). 9c

9d Though their functions overlap, the emphasis is different in WUC partly due to the tremendous differences in media. Since it lives in an on-line environment, and does not have the limitations of paper, WUC is meant to be more encompassing than WEC though computers certainly have their own limitations. The Whole Universe Catalog was conceived (before its name) to be more of a free access holistic education tool and on-line library, rather than an access to tools for an alternative life style for which WEC has developed a reputation. WEC is also concerned with self teaching and even begins with the "Understanding Whole Systems" section. To complete the overlap, WUC certainly has lots of room for providing access to tools for an alternative life style. 9d

9e What is copied directly from WEC to WUC are 1) the integrated, holistic way of viewing and comparing things, 2) providing alternatives to, and a co-evolution with, the existing system, and 3) the spirit behind the business honesty which resulted in the tradition of publishing an overview of the money accounts at the end of every issue. 9e

Thw Whole Universe Catalog: a way of looking at things

(J32782) 18-JUN-75 17:02;;;; Title: Author(s): Kirk E. Kelley/KIRK;
Sub-Collections: SRI-ARC; Clerk: KIRK;