79-0581

(1)

Semi Annual Technical Report on the National Software Works for the period 12 December 1978 through 30 June 1979

by

Massachusetts Computer Associates, Inc.

July 12, 1979

CADD-7907-1201

Massachusetts
Computer Associates, Inc.

MASSACHUSETTS COMPUTER ASSOCIATES, INC.
26 Princess Street
Wakefield, Massachusetts 01880

Semi Annual Technical Report on the National Software Works for the period 12 December 1978 through 30 June 1979

by
Massachusetts Computer Associates, Inc.

July 12, 1979
CADD-7907-1201

Effective Date of Contract:	12 December 1978
Contract Expiration Date:	12 December 1980
Amount of Contract:	$889,500.00
Contract Number:	F30602-79-C-0100
Principal Investigator and Phone Number:	Charles A. Muntz (617) 245-9540
Project Scientist or Engineer and Phone Number:	Richard Robinson (315) 330-7746
ARPA Order Number:	3686

This work was supported by the Advanced Research Projects Agency of the Department of Defense and by Rome Air Development Center. It was monitored by Rome Air Development Center under contract number F30602-79-C-0100.

TABLE OF CONTENTS

1. Introduction

The National Software Works (NSW) is a significant new step in the development of distributed processing systems and computer networks. NSW is an ambitious project to link a set of geographically distributed and diverse hosts with an operating system which appears as a single entity to a prospective user.

The National Software Works is being developed in response to a growing concern over the high cost of software. The Air Force has estimated that in FY72 it spent between $1 billion and $1.5 billion on software, about three times the annual expenditure on computer hardware. The Air Force has further estimated that by 1985 software expenditures will be over 90% of total computer system costs.

Since the early days of computing, in fact, the cost and complexity of developing and maintaining software have been substantial obstacles to the efficient and effective use of computers. To breach this barrier, both industry and government have committed vast resources for the development of tools -- automated aids for the implementors of software and the managers of software implementation projects. These tools include compilers editors, debuggers, design systems, test management tools, language analyzers, etc.

The difficulty is not the existence of suitable tools for a given programming task; it is the availability of the tools. The notion of software portability, often proposed as the solution for the problem of providing programming tools in some environment, has proven to be a will-o'-the-wisp which the industry has vainly pursued for the past twenty years.

The success of the Arpanet in providing programmers economical access to geographically dispersed computers provided the foundation on which the NSW concept was built. Instead of moving the software from host to host, let the programmer (and manager) use each software tool on whatever host it already occupies. To take a specific example, the Navy requires a programming support environment for the UYK-20 minicomputer. There currently exist cross-assemblers and compilers for the UYK-20 on IBM 360 hardware. On TENEX there is a UYK-20 emulator and debugger. MULTICS has the QEDX editor. All three of these host computers are connected by the Arpanet. Solution -- let the programmer use these existing tools to develop UYK-20 software.

That solution sounds plausible, but it ignores some serious practical considerations.

- You need an account on each host. This involves the allocation of funding, drawing up contracts, etc.

- The operating system on each host is different, so you must learn different login procedures, command languages, interrupt characters, file naming conventions, etc. Further you must not confuse each system's conventions as you move from tool to tool.

- Files output from one tool (say QEDX on MULTICS) are to be input to another tool (say CMS2M on IBM 360). This involves at least network transmission and usually file reformatting. To appreciate the magnitude of this problem one should try to use FTP (Arpanet File Transfer Protocol) to move a QEDX output file -- a sequential file of 9 bit ASCII characters in 36 bit words -- to an IBM 360 to be a CMS2M input file -- a blocked file of 80 EBCDIC character records in 32 bit words.

These and similar problems will be familiar to anyone who has used several different systems.

The purpose of NSW is to make this solution (of providing programmers access to tools on different hosts) a practical reality. The NSW user should not have to know about OS/360, TENEX, and MULTICS with their differing file systems, login procedures, system commands, etc.; knowledge of how to use the individual tools which are needed for the job should suffice. He should not have to worry about reformatting and moving files from a 360 to a TENEX; file transmission should be completely transparent. The user should not have to worry about obtaining accounts on many different machines, but instead should have a single NSW account.

Thus, the National Software Works is to provide programmers with a

- Unified tool kit - distributed over many hosts, and a

- Single monitor with
  - uniform command language,
  - global file system,
  - single access control, accounting, and auditing mechanism.

2. History of NSW Project

2.1 NSW Goals

2.1.1 Original Directions

As originally conceived, NSW was to provide the above-described facility in the context of certain specific external goals. The first such goal was large scale. Contemporary operating systems support tens of concurrent users. NSW was to support many more users, possibly as many as one thousand. The catalogue alone of the file system for that many users could easily fill a large disk pack. The table space required for keeping track of one thousand users and the software tools that they are using could easily exceed the virtual memory of TENEX.

The second goal was high reliability. If there are one thousand online users, then a two hour system failure costs one man-year of work. The National Software Works -- particularly its monitor and file system -- must degrade gracefully. Failure of a single component -- e.g., a TENEX system on which tools are running -- must only reduce system capacity, not destroy it. Further, only those users actually using a failed component should be affected by its failure.

The third goal was support of project management. NSW was to provide managers of software projects with a collection of programs, called management tools, which they can use to monitor and control project activities. The underlying assumption here is that a manager's ability to insure that each programmer's efforts contribute most effectively to overall project goals can be greatly enhanced by automating routine management tasks. Furthermore, it is assumed that a good environment for this automation is the system which supports the project programming activities because it represents an effective point for monitoring and controlling those activities.

The fourth goal of NSW was practicality. NSW was not to be a "blue sky" system, whose implementation required unrealistic assumptions about its environment. In particular, practicality meant:

- Minimum modifications to existing operating systems on Arpanet hosts. Minimum was, in fact, to be construed as none. It was possible to add privileged (i.e., non-user) code to the existing systems, but the solution to the problem should not depend on rewriting the kernel of any existing operating system.

- Minimum modifications to existing tools. Here, minimum no longer meant none. It was possible to require some change to a tool as part of the process of NSW installation, but such changes should be small scale and contained.

- Maximum generality. Any solution which permits the easy installation of existing tools must also allow the easy construction and installation of new tools.

- o No experimental hardware. This requirement meant that new hardware-oriented-approaches to reliability - e.g., PLURIBUS - cannot be used. The NSW monitor and file system are to run on already available Arpanet hosts.

2.1.2 Current Project Goals and "Immersion"

The original goal of supporting as many as one thousand users with a single NSW has been replaced with the objective of support of as many as one hundred users by several "regional" NSWs, distributed as a product and operated by local personnel. Rather than distributing the system data base and synchronizing the updating of the distributed data base with enriched protocols, the project is adopting the software practices that allow the system to be distributed in toto: configuration management, software error reporting/correcting protocols, etc.

Reliability issues are being addressed with the following question: If the central monitor was down, could tools still operate in some limited but useful context? Thus, rather than cast the monitor as cooperating distributed processes, allow tool execution to be insensitive to absence of the monitor. Of course the user's right to access the tool must have been previously established by the central monitor, as must the user's right to access certain files. Similarly, distribution of results requires presence of the monitor.

In general, NSW ought to be able to support its own development, distribution, and maintenance. This requirement generates quite a number of goals: support a product which is fabricated on diverse hardware by geographically dispersed contractors. This requires implementation and support of configuration management, use of "local" computing resources with (nearly) the ease of native operations, electronic mail, etc. Since the NSW project bears such resemblance to its potential uses, it serves as an excellent model to the system's developers. We have adopted the term "immersion" for this concept.

2.2 NSW Architecture

In this section, we summarize the NSW design, indicating what effect the NSW goals had on the system architecture. We can factor the NSW problem into two parts:

- o The development and implementation of methodology for excising tools from their current environments and interfacing them with the new NSW monitor.

- o The design and construction of a unified monitor and file system for the Arpanet.

In the next two subsections we examine each of these problems in turn and describe the components of NSW which provide solutions to the technical difficulties involved with each part.

2.2.1 Tool Kit

We first have the task of excising tools from their current operating environment and embedding them in the new one. In the context of the goals of NSW, we will discuss the technical issues which must be solved in order to provide the requisite tool installation methodology.

By its very definition, NSW is a distributed system. Tool processes run on different Arpanet hosts. The monitor must run on at least one Arpanet host. There must be some form of inter-host inter-process communication. There are low level Arpanet protocols for moving bits from host to host, and there are also several higher level protocols for moving files and for terminal communication. None of these protocols, however, is oriented toward the kind of inter-process communication which NSW requires. Moreover, even though NSW is being implemented on the Arpanet, we want to keep it as independent as possible of the underlying milieu. Network technology is evolving, and we wish to be able to realize the NSW architecture on tomorrow's networks as well. Hence, the first technical problem to be solved is the definition and implementation of an appropriate inter-host inter-process communication protocol. The protocol developed for NSW is called MSG.

The user of a tool has a variety of mechanisms for communicating with the tool. The user's terminal must be interfaced to the system and its peculiarities handled -- for example, the right amount of padding added after a carriage return. Control characters which happen to be meaningful to the local host must be intercepted before they reach the local executive. In order to allow uniform access to all the tools in NSW, running on many different machines, we must define a standard set of control functions and implement a system component which interfaces the user to every tool. The problem of standardizing control functions and insulating the user from the vagaries of the different operating systems is handled by an NSW component called the Front End.

A tool running on some machine makes system calls requesting resources -- primarily file access. Since access to NSW system resources is to be controlled, accounted for, and audited by the NSW monitor, such requests must be diverted from the local system and instead referred to that monitor. In addition, if the tool is interactive, it expects to have a terminal for communication with the user, and this in NSW is via the Front End. So, without modifying the operating system, we must divert the tool's communications with the user and the tool's requests for local resources. The NSW component which solves this problem is called the Foreman.

Batch tools are best described as those whose input can be entirely specified before tool execution begins. Such tools should not (and often can not) be supervised from a terminal. Rather the central monitor works together with a component called the Batch Job Package, running on the same host as the batch tool, to supervise execution of such "absentee" computations.

Finally, we expect that the output of one tool will be used as input to another tool. Unfortunately, if the first tool is a MULTICS editor and the second an IBM 360 compiler, this operation involves character translation (ASCII to EBCDIC), file reformatting (sequential file to blocked, recorded file), and file movement (across the Arpanet). To handle such file transformations and movements there is an NSW component called the File Package.

It is worth noting at this point that all of the above components are distributed. Every host in NSW has an MSG server process. Every site to which a user is connected has a Front End. Every tool bearing host has a Foreman. Every host on which NSW files are stored has a File Package. It is also worth noting that implementation details of these components vary from host to host. A MULTICS Foreman will be vastly different from an IBM 360 Foreman. Functional specifications for these components are fixed throughout NSW, but implementation and optimization decisions are left free.

Before proceeding to the NSW monitor, let us summarize the technical problems and the resulting components which provide the unified tool kit methodology.

o Inter-host inter-process communication	MSG
o User interface	Front End
o Diversion of communication with local operating system	Foreman
o Supervision of batch jobs	Batch Job Package
o File transformation and movement	File Package

2.2.2 NSW Monitor and File System

The design of the NSW Monitor - called the Works Manager - was probably more affected than any other component by the goals of NSW. Functionally it is not different from any other conventional access-checking, resource-granting monitor. Structurally, however, it is significantly different.

The goals of providing both large scale and reliability on conventional hardware led to the approach of distributing the Works Manager and file system. If there are many instances of the NSW monitor on many different hosts, then failure of a host is not catastrophic. Unfortunately, distribution runs counter to the problem-required logical unity of the monitor and file system. If a user inserts a file into the file system using one tool and one instance of the file system, and then requests the same file using a different tool and a different instance of the file system, the two instances of the file system must share a common file catalogue for the system to behave properly. Similarly, all instances of the monitor must share an access rights data base for proper validation of user requests to run tools.

One solution is to partition the Works Manager database and distribute the partitioned "pieces" so that the Works Manager on each host can allocate resources that it "owns" directly, but must negotiate with the Works Manager that "owns" other resources. This strategy requires minimum synchronization while providing advantages in reliability and robustness.

2.3 Phases of NSW Development

The design and implementation of the National Software Works has proceeded in four slightly overlapping phases

- Structural design and feasibility demonstration
- Detailed component design
- Prototype implementation
- Reliability and performance improvement

In the following subsections we describe these phases in more detail.

2.3.1 Structural Design and Feasibility Demonstration

The first phase of NSW development began in July, 1974 and concluded in November, 1975. During this period, the basic architecture of NSW (described in Section 2.2) was established. Further, relatively ad hoc implementations of major components were made. These components were integrated into a system which was demonstrated to ARPA and Air Force personnel at Gunter AFB in November, 1975. This demonstration exhibited various system functions, the use of batch tools on the IBM 360 and Burroughs B4700, the use of interactive tools on TENEX, transparent file motion and translation, and a primitive set of project management functions.

This demonstration confirmed that the expected NSW facilities could be implemented and that transparent use of a distributed tool kit was feasible. The NSW System, however, was inefficient and fragile. Further, many of the ad hoc implementations had design weaknesses which limited their general application to a sufficiently broad range of hosts and capabilities. For these reasons, an effort was begun to produce adequate component designs.

2.3.2 Detailed Component Design

This second phase of NSW development was begun in June, 1975 with the initial MSG design document. Specifications were developed for Tool Bearing Host components - MSG, Foreman, and File Package. All of these specification documents were completed by March, 1976. (They have all been revised since then, but the original specifications are still substantially correct.)

During the same period, the external specification of the Works Manager was also made. Again, although this specification has subsequently been revised, it is still substantially correct. The remaining portions of the core of NSW - i.e., the batch tool facility: Works Manager Operator, Interactive Batch Specifier, and Interface Protocol - were designed during phase one, and those designs were retained until phase four (see below).

The remaining major NSW component, the Front End, was the subject of several design efforts. Three incomplete specification documents were produced but none of these was wholly satisfactory. Nevertheless, sufficient design to allow implementation of a functionally correct Front End was accomplished. Completion of a general specification for the Front End is one of the tasks remaining to be accomplished.

2.3.3 Prototype Implementation

As specification documents were completed, various contractors began implementation of the NSW components on the initial set of hosts - TENEX, MULTICS, and IBM 360. These efforts commenced in January, 1976. Implementation on TENEX proceeded more quickly than the efforts on the other hosts - primarily because the MSG system designers were also TENEX implementors. By October, 1976 prototype implementations which conformed to the published specifications had been made for all TENEX TBH components. In addition, all components of the core system were available on TENEX.

Implementation of TBH components on MULTICS and IBM 360 proceeded more slowly; however, initial implementations of MSG components on both of these hosts were completed by the end of 1976. By November, 1976 sufficient progress had been made on implementation of a File Package and Foreman on MULTICS that it was possible to demonstrate an interactive tool running on MULTICS. Progress on implementation of 360 (interactive) TBH components reached a similar position in September, 1977.

Also during this phase, a TENEX Front End which functionally supported the Works Manager and Foreman according to the appropriate specifications was implemented.

An NSW system containing prototype implementations according to the specifications of the core system, TENEX TBH components, TENEX Front End, batch IBM 360 tools, as well as a rudimentary MULTICS interactive tool was demonstrated to Air Force and ARPA personnel in November, 1976. At the same time, a demonstration of MSG components on all three hosts was also given.

2.3.4 Reliability and Performance Improvement

Even though implementation of components on MULTICS and IBM 360 was lagging, implementation of the core system, TENEX TBH components, and TENEX Front End had proceeded to the point that the issues of reliability and performance assumed major importance. The system exhibited sufficient functional capability that it could clearly support use by programmers if it were sufficiently robust and responsive.

The first task attacked was to provide robustness. Work had begun on a full-scale NSW reliability plan in 1975. The detailed plan was released in January, 1977. Since it was clear that implementation of the full plan was a major undertaking, a less ambitious interim reliability plan which ensured against loss of a user's files was begun in mid-1976. This plan was also released in January, 1977. By June, 1977 the core system, TENEX Foreman, and TENEX Front End had been modified to incorporate the features of that interim plan. In addition, both the MULTICS and IBM 360 Foremen (only partially implemented) were altered to conform externally to the scenarios specified by the interim reliability plan. A system exhibiting the new scenarios was released for use in June, 1977.

Performance of NSW had been slow from the initial implementation. The reasons for slow response were many:

- o interaction between components was by a thin wire (MSG and the Arpanet).
- o NSW components (which constitute an operating system) nevertheless were executed as user processes under the local host operating system.
- o Component implementation had been oriented towards ease of debugging and other concerns of prototype systems rather than towards the performance expected of a production system.

In 1977, efforts to improve NSW performance were begun.

The first effort was the development of a performance measuring package for TENEX MSG. Results of the first set of measurements were reported in April, 1977. Some performance improvements were suggested by the initial measurements, but the most obvious suggestion was that more sophisticated measuring packages were needed. Several such packages were begun to perform various kinds of measurements on TENEX components. All of these packages were complete by February, 1978. By May, 1978, all TENEX components had been instrumented and measurements of page use, CPU time, elapsed time, use of JSYS (TENEX system commands), etc. had been taken under a variety of system load conditions and on several different TENEX hosts. Efforts are currently under way to implement the performance improvements suggested by these measurements. Performance improvements have already been made to several components. Results of these improvements are described in section 3.2 below.

Concurrent with the effort to improve NSW reliability and performance, an effort to make NSW a more packaged product were begun. Regression tests for the externally available NSW user system were developed and applied to each system release. A user's manual for the system was published. Documentation of the core system was produced. Finally, a draft configuration management plan was developed.

Phase four of NSW development is still continuing. Efforts to improve performance of TENEX components are substantially complete. Certain features of the full scale reliability plan have also been implemented, and phase four should be complete by mid 1979. Phase five, development of a production NSW system, is underway. The efforts proposed for phase five are described in section 4 below.

2.3.5 Production System

Work was begun in late 1978 to establish NSW as a software product. The NSW Management Plan was generated. This document identified a number of roles associated with development, operation, and support of NSW as a product. Briefly these are as follows:

Role	Responsibilities	Organization
Policy Group (PG)	Requirements and Policy	RADC/ARPA
Product Development (PDC)	Product Definition	GSG
NSW Operations (OPS)	Operations; User Support	GSG
Architecture Control (ACC)	Product Integration	COMPASS
Development and Maintenance (DMC)	Software Development and Maintenance	BBN, COMPASS, HIS, UCLA
Tool Manager (TM)	Tool Management	(To be announced)

A product baseline is being established to bring NSW under Configuration Management. A reasonably complete set of requirements/specification documents is being installed as a set of files in the NSW User System. NSW's Information Retrieval System allows queries on sets of documents; the naming scheme for the baseline demonstrates some of the power of NSW's file naming capabilities:

Document Type	Name Syntax
Requirements	NSW.REQUIREMENTS...
A-level Specifications	NSW.A-SPEC...
B-level Specifications for <component>	NSW.B-SPEC.<component>...
C-level Specifications for <component> on <operating-system>	NSW.C-SPEC.<operating-system>.<component>...

where the variables take on values as follows:

<component>	<operating-system>
BJP	TENEX
FE	OS360
FL	MULTICS
FM	UNIX
FP	...
...	

Revision level of all of these documents is also noted, but not yet in final NSW form.

A method of tracking the progress of bugs and improvements has been established. Software Trouble Reports (STRs) and the more general NSW Standard Transactions (NSTs) have been introduced as (essentially) numbered entries in a product development journal. Whenever a new bug or suggested improvment is reported, it is assigned the next available journal entry and recorded.

A protocol for correcting bugs and implementing suggested improvements was outlined in the NSW Mangement Plan. The objective of the protocol is to define a system in which NSW Product Oganizations (PDC, ACC, DMCs, etc.) receive STRs and NSTs, perform their designated function, and then forward the revised transaction accordingly. Much of this "transaction processing" is currently being conducted via more-or-less structured use of the ARPAnet mail system. A more orderly approach to this problem is under consideration: a modest tool, christened MONSTR (Monitor STRs), would actually implement the management plan by dealing directly with each participating organization according to desired protocol.

Since many of these capabilities are still evolving, none of this work has led to new system functionality. A long-term goal of the project - especially visible in the context of immersion - is to move towards machine-based configuration management. Thus, if NSW system release is actually performed by releasing a (large) hierarchy of files - of sources, objects, executables, documents, etc. - system primitives should eventually automate the record-keeping aspects: lists of files with appropriate revision level, lists of files changed since the previous release, etc.

Release 4.0 of NSW was delivered to PDC on May 1, 1979. It was chiefly a maintenance release, but some new functionality was included:

- a new component, called the Fault Logger (FL), is available to collect error messages, log them, and forward them to operations and system personnel.

- a system tailoring facility was provided by defining the Configuration Database. This text database is identically present on each NSW host; each host can read the file to determine local parameters as well as a description of remote NSW resources. All Tenex and TOPS-20 hosts read this file at initialization; other hosts will behave uniformly in this regard by late 1979.

- operator's utilities have been greatly improved, although more work needs to be done. A comprehensive Release Specific Document was produced, detailing the systems component parts, integration constraints, and operating considerations.

3. Current Status

3.1 Overview

The NSW system currently available to users is NSW 3.1, released in November, 1978. A new version, release 4.0, has been undergoing acceptance testing since May 1, 1979. This release was a maintenance release which featured improved operating characteristics and a more carefully controlled release procedures. The features of NSW 4.0 are:

- The same basic features as NSW 3.1, i.e.:
  - Twenty interactive TENEX tools, many of which are available on TOPS-20.
  - Ten interactive MULTICS tools.
  - One interactive IBM 360 tool, and nine IBM 360 batch tools.
  - Basic set of system commands.
  - User documentation and support.
  - Rudimentary management (node manipulation) tools.
- And the following extensions and changes:
  - Configuration of NSW 4.0 includes the following hosts:

    ISIE
    ISIC
    RADC-20
    CCN-360/91
    RADC-MULTICS

  - The management/node manipulation tools now function in a manner similar to the set-up operations for batch tools - interactive, but not true, suspendable, tool-instances - and are controlled by the tool rights mechanism.
  - The implementation of the Works Manager file attribute facility allows installation of new 360/91 interactive tools.

- o Significant operational improvements have been made, particularly:
  - o creation of release-specific documentation
  - o implementation of a configuration control facility to handle site-specific parameters
  - o some rationalization of procedures
- o The release procedure has been significantly formalized, and brought more under control. Changes to NSW 4.0 core system over the life of the r·lease by maintenance of a source code repository.
- o A new component, the centralized Fault Logger, has been added.
- o Support of a UNIX Front End has been anticipated.
- o A formal system for handling trouble reports and improvement requests has been developed.

Functionally, the current NSW system is minimally adequate. It has a reasonable collection of tools, but many of these tools have not been adequately tested. The minimal set of user commands is available and tested, but many needed user features are lacking - e.g. command macros, '*' in file commands, I/O devices, Arpanet mail, etc. Performance has been improved significantly. The documentation of system components has been improved, but much needs to be done.

TENEX and TOPS20 are available as Works Manager or Tool Bearing Hosts according to specification, but TOPS20 tool encapsulation is currently less satisfactory than TENEX. Additional encapsulated tools can be installed in either environment to increase NSW capacity. Batch tools are available on the CCN IBM 360/91, and more can be installed as needed. A major overhaul of the entire batch system has made it more consistent with the rest of NSW, more flexible, powerful, operable and resilient. The IBM 360 Foreman implements only one interactive tool, and a minimal set of specified features. The MULTICS implementation has been improved greatly over NSW 3.1, although problems persist - particularly in the Foreman implementation.

The current status of the individual component implementations is presented in section 3, and planned improvements to the system are presented in section 4.

3.2 Components

In the following subsections we give a description of the current status of each NSW component.

3.2.1 Core System Components

The core system components - Works Manager, Checkpointer, and Works Manager Operator - are substantially complete. The Works Manager has been the object of an extensive and successful effort to improve its performance. The Checkpointer has had its functionality enhanced, and been made more robust. The Works Manager Operator has been substantially rewritten to interface to the Batch Job Package, and to conform to the coding standards imposed on the Works Manager.

3.2.1.1 Works Manager

At present, the Works Manager consists of a number of identical concurrent instances of the same program, each one working on a single request at a time. All such processes share two common data bases, the Works Manager Table data base and the NSW File Catalogu . In addition to these processes there is a separate process, the Checkpointer, which makes periodic backup copies of the data bases.

The Works Manager supports 31 different Works Manager procedure calls, which are available to other NSW processes. These procedures are described in the Works Manager System/Subsystem Specification and the Works Manager Program Maintenance Manual.

A substantial effort was invested in implementing the scenarios described in the "Interim NSW Reliability Plan" (CA-7701-2111). These scenarios are as close as possible to the final NSW design which is described in "NSW Reliability Plan" (CA-7701-1411). The goal of these scenarios was to guarantee a user that a system malfunction -- other than catastrophic disk failure -- would cause few, if any, of her/her files to be lost. This guarantee includes files stored in the NSW file system as well as closed local files in a tool's workspace. It was not a goal to provide continuity of service in the face of individual component failure, nor was it a goal to eliminate long (possibly endless) waits by the user in the event of message delays or component failure (these desirable goals would be met by implementing the complete reliability plan).

In order to guarantee that NSW file system files not be lost (except under rare circumstances) it was necessary to preserve the NSW file catalogue. It was presumed that these files themselves are preserved by some mechanism on the file bearing host. Periodically (currently at approximate twenty minute intervals) the WM file catalogue is locked, the entire file catalogue is copied onto disk, and then the lock is released. The WM also maintains a data base of active users, active tools, etc., which is also copied onto disk (using the same mechanism described above for the catalogue). The Checkpointer, a new NSW component, was designed and implemented to fulfill this function.

The twenty minute interval introduced a window during which a file transaction may be lost if the WM host should crash. This twenty minute interval is sufficient with respect to NSW Exec commands. However, a tool might wait until termination to deliver any files; in this case, many hours of work could be lost. In order to avoid this problem, a mechanism was developed so that a Foreman could ensure the preservation of the local tool workspace (LND) in the event of either local host crash or the failure of other NSW components. The LND contains any files being delivered by the Foreman on behalf of the tool.

The mechanism developed ensured that the LND is preserved until after a file catalogue containing references to delivered files has been checkpointed. The LND is only (intentionally) erased after tool termination. Whenever a tool terminates normally, an additional message (FM-GUARANTEE) is sent by the Checkpointer (the process performing the file catalogue checkpoint) to every Foreman instance which terminated since the last checkpoint. Each Foreman instance sets a timer and if the FM-GUARANTEE message is not received when the timer goes off, the Foreman saves the LND.

The requirement for the Foreman is that it must be able to maintain the LND is such a way that it is preserved over Foreman host crashes. The Foreman must be able to explicitly invoke this save-the-LND mechanism. This allows the Foreman to explicitly preserve the tool's workspace should any difficulties arise during some scenario.

The AUTOLOGOUT scenario is initiated by a break in the connection between the user's terminal and the Front End. All running tools are forced to stop and initiate the save-the-LND mechanism described above.

A mechanism was also implemented which allows the user to have (some of) the saved files delivered to the NSW file system. This mechanism is provided by the LNDSAVED and RERUNTOOL sceanarios. Once a Foreman has performed the save-the-LND mechanism, it informs the Works Manager. The Works Manager maintains a record of such saved LNDs in each user's node record. A message will be sent to the user at each subsequent login until the user causes its deletion by using the RESUME command (which invokes the RERUNTOOL scenario). The user will receive messages about the saved LND until the user explicitly saves the files (TERMINATE subcommand) or deletes them (ABORT) subcommand). Currently, these are the only two options of RESUME which are implemented; it has been proposed that RESUME be expanded to allow the user to restart an instance of the same tool in the saved LND.

The management/node manipulation tools are implemented entirely within the Works Manager. These are now invoked under the tool rights mechanism using the same interactive/HELP technique as batch tools. This has allowed removal of the specialized Front End/Works Manager interface formerly used, eliminating five special Works Manager procedure calls, and eliminating all knowledge of these tools from the Front End. Thus the UNIX Front End development is also freed of this knowledge.

The file attribute mechanism was also implemented, allowing Foreman calls to the Works Manager for getting and delivering user files to specify the file type. This feature was required to support future 360/91 interactive tool installation.

The Works Manager also uses the Global Configuration File (see 3.2.1.6), which has given it more operational flexibility. In particular, its timeouts on calls to remote procedures can be tuned without affecting other components. Also, event logging can be more flexibly specified, and better accomodates the divergent needs of the developers and operators.

The Works Manager, which consists of approximately 25.7K lines of BCPL code, is structured into a number of layers. At the top level, WMMAIN waits for a procedure call message from another NSW process, does initial decoding and validity checking of any such message, then dispatches the message to the proper routine. The Works Manager Routines, WMRTNS, implement the 31 Works Manager Procedures. At their disposal are a number of lower-level utility packages and subsystems. The Works Manager Table Package, WMTPKG, handles all interactions with Works Manager tables. It serves as an interface to the Information Retrieval System, INFRTV, which manages the NSW File Catalogue and the Works Manager Tables. All NSW processes written in BCPL have available NSUPKG and BCPPKG. NSUPKG contains a number of facilities to handle MSG messages, create and record NSW fault descriptions, etc. BCPPKG provides basic utilities to handle character strings, do searching and sorting, and so forth.

As with other core system components and the TENEX/TOPS20 File Package, the Works Manager is transportable between TENEX and TOPS20 without modification.

3.2.1.2 Checkpointer

The Checkpointer status mimics that of the Works Manager, since it consists largely of the entire Works Manager utility package, with a relatively small upper layer of code to implement the specific Checkpointer procedures. Thus, like other core system components, the Checkpointer is transportable between TENEX and TOPS20 without modification. The performance improvements realized by the Works Manager table Facility also apply to some Checkpointer procedures.

The Checkpointer has the following characteristics:

- o Implements the FM-GUARANTEE call on the Foreman required by the Interim Reliability Scenarios.
- o Manages NSW file deletion. Files deleted by the user are actually deleted by the Checkpointer after a time interval, as required by the Interim Reliability Plan.
- o Makes Checkpoint files of all Works Manager database files at configuration controlled intervals.
- o Is robust and flexible to about the same level as the Works Manager itself.

The Checkpointer received a major re-write for NSW 4.0. A completely new asynchronous remote procedure call handler was written. This allows the Checkpointer to make multiple simultaneous remote procedure calls/usually to delete files without interfering with the timing of database checkpoints.

The Checkpointer also uses the Global Configuration Database file, and has gained significant flexibility as a result. The external procedure call timeout, checkpoint interval, and waiting period before file deletion occurs are all under operator control.

The Checkpointer is halted by an interrupt from the operator utility OPRUTL (3.2.1.4).

3.2.1.3 Works Manager Operator

The Works Manager Operator has been extensively used with the 360/91 Batch Job Package since November 1978. Detail improvements have been made to improve reliability and job status reporting.

WMO shares a data base (the Job Queue File) with the Interacive Batch Specifier (IBS) module in the Works manager. We intend to remove this shared access by making all access to this data base be via procedure calls on WMO, which will have sole access. To this end, direct access to the data base by the WM to get a batch job status (NSW: JOB) has been replaced by a call on WMO by WM on the (new) WMO-SHOWJOB procedure. Direct access to the data base by IBS will be replaced by use of a WMO procedure, WMO-ENTERJOB, to be specified and implemented in the future.

WMO also uses the configuration database file.

Some notable characteristics of the current WMO are as follows:

- WMO is responsible for both processing the Job Queue File and handling WMO procedure calls. These two tasks are handled by distinct instances of WMO in any given NSW system.

  (1) There is exactly one instance of WMO processing the job queues. A standard locking discipline guarantees that precisely one such instance exists. This instance executes the job steps necessary to process a batch job, and initiates all procedure calls to external processes (WM, BJP, FP). It never receives generically addressed MSG messages.

  (2) There are zero or more instances of WMO which receive generically addressed MSG messages, and handle all currently defined WMO procedures. These instances never execute job steps or initiate external procedure calls. Thus, these instance(s) provide external access to the data base.

- A primitive retry mechanism exists. WMO will retry an external procedure call indefinitely when it fails due to network or remote host crash. It will retry a failed external procedure call a maximum of three times if the failure is due to resource problems, e.g. no disk space.

- Status reports generated by WMO for display by WM (NSW: JOB) have been made more informative; all information supplied by BJP is reported.

- The maximum number of jobs in the job queue file is currently 64. This may be increased when needed, but requires re-compilation and reloading of WMO.

- The WMO cycle number may be set manually by the WMO utility (WMOUTL), but does not automatically increment with each cold start. "Cold Start" in this version occurs only when a ne new job queue file is created.

3.2.1.4 Operator Utility - OPRUTL

The opertor utility program, OPRUTL, has become sophisticated enough to be mentioned as a core system component in its own right. It can operate either stand-alone or as an actual NSW component under MSG. It allows the NSW to perform some maintenance functions which are tedious with more primitive developer-oriented utilities. Its capabilities are:

- To clean all or specified LOGIN entries out of the Works Manager database, e.g., to recover from a core system host crash.
- To enter new tool descriptors into the Works Manager database. This is the only pratical means of entering batch tools, which must be parsed and error-checked on entry.
- To stop the Checkpointer via an MSG alarm.
- To report on current NSW usage, i.e., who is logged in.
- To reset all internal database locks (as a cleanup operations).

3.2.1.5 Central Fault Logger (FL)

NSW 4.0 contains a prototype implementation of a centralized fault logging component. This component is intended largely to be an operational aid, by providing a single component through which the operator can receive fault messages from any component in the system - remote or local.

The Fault Logger design provides for sophisticated facilities to filter incoming messages and route them to several alternative or multiple destinations - devices, terminals, or files. The prototype FL simply maintains a fault message database in Arpanet mail file format, allowing mail handling tools such as HERMES to be used for information retrieval.

The only component which currently logs faults to the FL is the TENEX/TOPS20 Foreman. The next NSW release will see expanded use of a more sophisticated FL.

3.2.1.6 Global Configuration Database

NSW 4.0 includes initial use of a prototype configuration database which supports specification and control of site dependent configuration data. It consists of a specially formatted text file containing parameters which apply both to a whole NSW configuration - hosts used, MSG generic classes defined, etc - and to a specific host in the configuration - directories used, timeout values, logging parameters, etc. The database is designed to be maintained at a central site and then be broadcast to all hosts in a given configuration, giving NSW operations a centralized means of controlling an entire NSW configuration.

The current database is a prototype used by most of the core system components and the TENEX/TOPS20 File Package. Other components are expected to implement use of ths database in future releases. Configuration data now used include:

- WM - system herald, remote call timeout, event logging.
- CHKPTR - remote call timeout, deleted file wait interval, checkpoint interval, event logging.
- WMO - remote call timeout, event logging.
- FLPKG - remote call timeout, event logging, filespace directory name.

3.2.2 TENEX/TOPS20 TBH Components

The TENEX/TOPS20 TBH is the most advanced of the three TBHs. All components (MSG, Foreman, and File Package) are substantially complete and tested. All components are transportable between TENEX and TOPS20.

3.2.2.1 MSG

The MSG specification was produced in January, 1976. It was revised in December, 1976 - primarily to resolve ambiguities in the earlier document. It was extended in April, 1978 to allow for support of multiple, concurrent NSW systems. The TENEX/TOPS-20 MSG component implements the revised and extended specification with only two exceptions (which are noted below).

The TENEX/TOPS-20 implementation of MSG is a single executable module which will run under TENEX, TOPS-20 Version 101B, and TOPS-20 Release 3. In addition to the communication functions supported for processes (and defined by the MSG-process interface specification) the TENEX/TOPS-20 implementation includes a powerful process monitoring and debugging facility, and comprehensive performance monitoring software.

The TENEX/TOPS-20 implementation does not perform MSG-MSG authentication. Message sequenceing and stream marking are not implemented (however the underlying software structure exists to support both).

The current implementation was extended to support new component initiation features required to support TOPS20 TBH components. In addition, a recent modification to MSG supports rapid timeout of attempts to contact remote hosts where an MSG is not up, or which are themselves down. This markedly reduces the wait time imposed on a user who has attempted to use an unavailable resource.

The implementation has also been modified to enhance its performance, based on extensive performance measurements completed this year. Changes include elimination of network connections for local message traffic, data re-structuring, reduction of calls on expensive JSYSES, and improved strategies for memory allocation.

3.2.2.2 Foreman

The current TENEX/TOPS-20 Foreman (Version 1521) implements all scenario functions defined by the interim NSW reliability plan in its most recent revision (March 1, 1977). The Foreman only supports tools which run in encapsulated mode. It does not yet support the direct use of NSW functions by any class of tools. It currently supports approximately twenty TENEX and five TOPS20 tools in this encapsulated mode. Some of these tools have been extensively tested and used within NSW; others have merely been superficially exercised.

The latest release can operate on both TENEX and TOPS-20 Release 3 configurations. There is a single .SAV file which detects at runtime the configuration type and modifies its behavior accordingly. This newest release has now had adequate field testing on the TOPS-20 machines. Not all TENEX NSW tools are available on TOPS-20 and those that are have not been tested to the same degree as their TENEX counterparts.

The current Foreman implementation handles the problem of storing "saved" tool workspaces through the temporary means of utilizing the workspaces themselves. A permanent facility to handle workspace management is already designed and implementation is pending.

The TENEX/TOPS20 Foreman has been extensively modified as a result of the extensive performance measurements made in early 1978 and reported in BBN report No. 3847, "A Performance Investigation of the National Software Works System". Performance enhancement has been currently limited to reducing resource consumption by the Foreman e.g. by minimizing use of expenive JSYSes, pre-allocating workspace directories, etc. Future work will address alternative system support configurations, and altered patterns of NSW communications.

The NSW 4.0 Foreman incorporates improved reporting of user file delivery from tools, and reports faults to the Fault Logger.

3.2.2.3 File Package

The TENEX/TOPS20 File Package is now functionally complete. The task of writing Intermediate Language encode/decode for non-TENEX binary format files is now complete, and has been tested with the CCN/360 File package for several representative binary file types. The current File package version has the following characteristics.

- All specified File Package procedures are implemented and tested for local, family, and non-family network transfers. Unspecified procedures to support the obsolete IP mechanisms in WMO have been expunged.

- The Intermediate Language (IL) encode/decode package has been re-structured for greater efficiency and maintainability. Encode/decode has been partitioned into three classes - text files, sequenced text files, and binary files; there is an encode and a decode module for each class, totalling six. Code size has increased, but both efficiency and code comprehensibility have been greatly enhanced. The interface between the (BCPL) calling routines and the (MACRO10/20) service routines has been simplified. Impleme~tation of binary file encode/decode is complete, and has been extensively tested both against itself (i.e. against a remote TENEX simulating a non-TENEX host), and against the CCN/360 File Package. We have confirmed correct transmission of CMS2M object files from CCN/360 to TENEX/TOPS20.

- Performance enhancements have been implemented based on the results of BBN's performance investigation as reported in BBN report No. 3847, "A Performance Investigation of the National Software Works System", DRAFT VERSION, July 1978 by Richard E. Schantz. We have minimized the use of expensive JSYSes, notably the CNDIR (connect to directory) JSYS (average cost 220 ms per call). We have done so by specifying that the File Package must be able to create/read/delete files in its own filespace and Foreman workspaces without connecting to them, and letting it stay connected to its LOGIN directory. This has had no practical effect on the operation of NSW, beyond requiring that these directories be accessible from the system LOGIN directory. These enhancements hae resulted in a CPU usage reduction of up to 60% for delivery of a file from the Foreman workspace.

- The File Package is completely transportable between TENEX and TOPS20, requiring no modifications or patches. The simple transportability is based on the use of the Global Tailoring File for filespace name, logging information, and the use of the JSYS encapsulation packages now included in the Works Manager utilities. (See Appendix C).

- The logging of messages sent/received via MSG is under control of a spec in the Configuration Database (as in WM, WMO and CHKPTR). When logging is disabled, CPU usage for typical FP calls is reduced 25% - 40%. For comparison, the FP retrieval calls analyzed in BBN report No. 3847, "A Performance Investigation of the National Software Works System", DRAFT VERSION, July 1978 by Richard E. Schantz,. which averaged about 2.9 seconds, can be reduced to as low as 0.7 seconds with logging disabled.

The File Package is written primarily in BCPL (approximately 6.9K statements including utilities.) The IL encode/decode package is written in Macro-10 and consists of approximately 1.7K instructions.

3.2.3 IBM 360 TBH Components

The IBM 360 TBH is the second most advanced host. MSG and the File Package are substantially complete. The Batch Job Package is debugged and available. The weakest component is the Foreman which implements only a small subset of the specification.

A new overlay mechanism which supports overlaying of exclusive segments has been constructed and installed in the File Package, Foreman, and Batch Job Package. This mechanism was required to allow these components to fit in available real core, and to allow for incremental increases in code size.

3.2.3.1 MSG

The IBM 360 MSG component implements substantially all of the revised MSG specification. It does not yet implement the April, 1978 extension. The features of the current version are:

- Flow control is implemented for both sides.

- The present TENEX limitation of 2048 bytes per message is larger than CCN can handle reliably with its current allocation of resources to the NCP region. Therefore, CCN's MSG is being configured with a maximum inter-MSG message size of 1024 bytes.

- An MSG process can be materialized automatically in either TSO or batch. The IBM 360 MSG requires that a process specifically "materialize" itself with a system call to the central MSG. Included in this materialization call is an event signal which will be signalled to perform the "termination signal" function; however, at present MSG-central never signals this event. No mechanism exists to allow a process which is restarting after it crashed (while MSG-central stayed up) to resume its earlier instance number.

- Both Sequencing and Stream Marking have been implemented.

- MSG now includes the ability to automatically start a process under TSO when MSG initializes itself after a system crash.

- Authentication is implemented in a manner which does not match the current specifications. The most important difference is that an ICP is required to the CCN authentication socket.

- Binary direct connections may use any byte size, but byte sizes smaller than 8 bits are likely to lead to problems in determining the actual length of the message.

- It has been decided to provide for a manifold of coexisting NSW systems on the same ARPANET hosts. This requires that a host support multiple MSG's, using different contact sockets. The 360 MSG was implemented to allow both a "production" and a "test" MSG to coexists, using different contact sockets.
  It is planned to modify MSG to allow more than two different MSG's to coexist; this modification is not as trivial as it was once believed to be.

- The current process interface for direct connections blocks internally, so that the process does not receive control from an alarm until all direct connection I/O completes. The direct-connection interface must be changed to be non-blocking.

- Now optimizes the number of idle server processes maintained based on predicted system load.

3.2.3.2 Foreman

The IBM 360 Foreman provides only a subset of the features defined in the specification, as only features required to support the DISPLAY tool are implemented. Specifically:

- The 360 Foreman supports encapsulated tools only; in particular, there is no Foreman-tool interface.

- Encapsulation does not extend to the file system. Therefore, NSW files can be fetched only before the tool starts. Files cannot be delivered, as this feature is not required by DISPLAY. This is accomplished by the Foreman interpreting a control stream which it receives in the "filename-list" field of the FM-BEGINTOOL command. A tool cannot dynamically select an NSW file.

- The only tool-control command implemented is FM-BEGINTOOL; FM-STARTTOOL and FM-STOPTOOL are not implemented. Any non-zero value for Entvec is interpreted as 1, i.e, it starts the tool at the beginning.

- There is no Local Name Dictionary (LND), and hence no saving of LND's. FM-OK is not implemented. No LND cleanup process is started automatically after a system crash. FM-REBEGINTOOL is currently implemented as another name for FM-BEGINTOOL. Otherwise, tool starting and stopping follow the interim reliability scenarios.

- Implementation of the Works Manager file attribute mechanism will allow installation of new interactive tools.

3.2.3.3 File Package

The IBM 360 File Package implements substantially all of the revised specification. A few features have either not been implemented or have been incorrectly implemented. Specifically:

- All format effectors and record control tokens of IL are implemented. However, the variable format effectors HT, VT, LF, and FF, whose interpretations are defined for each file by the GFD are not fully tested with the Tenex File Package.

- The IBM 360 File Package never arms itself for alarms, and it never sends an alarm. If an error condition is found during data transfer, the IBM 360 File Package will immediately close the connection (rather than send an alarm, as called for in the specifications). The File Package has no mechanism for reporting the status of a transfer operation.

- The full Error Descriptors are not supplied by the File Package, due to PL/PCP restrictions. In particular:

  - The list of debug reports is always empty.
  - Only one error can be reported, the first one detected.
  - The values of the fault class and fault number fields have not been properly correlated with other File Package implementations.
  - The implementation of the Smithsonian Astronomical Date Standard is untested.

- A format for family copies of files which cannot be described in IL has not been defined or implemented for the IBM 360 family. Hence, all net transmissions, regardless of family, use IL.

- A local data set can be accessed by the File Package only if it exists within a directory in the NSW directory-group (i.e., having the NSW charge number). Since there is no mechanism to "connect" to a non-NSW directory, the password parameter is ignored.

- IL reblocking is not supported; a request to send an IL-encoded file with a transmission block size smaller than the IL blocksize in which it is recorded on disk may fail. This is not expected to be a problem, since File Package transmission block/sizes are expected to be established by gentleman's agreement and not varied.

- Binary I.L. encode/decode has now been tested and debugged with the TENEX/TOPS20 File Package.

- Only byte size 8 is supported for data transfer.

3.2.3.4 Batch Job Package

The initial implementation of the CCN/360 Batch Job Package is complete, and was released as a component of the candidate user NSW system on November 16, 1978. This implementation completely supports all BJP procedures specified in the revised Batch Job Package specification included as Appendix B to this report. This implementation has been extensively tested with the corresponding WMO version released on the same date, and has no known outstanding deficiencies. There are currently seven batch tools installed in NSW which may be run by WMO-BJP. Only the FORTRAN tool has been extensively tested and is known to run and produce good output. This testing deficiency is largely due to the circumstance that the personnel responsible for testing WMO-BJP are too unfamiliar with the other tools to create test input for them.

3.2.4 MULTICS TBH Components

The MULTICS TBH remains the weakest part of NSW. The components were implemented to comply only superficially with the specifications. The TBH components have been analyzed to a procedure Level, and a preliminary conformance study has been written. Problems have been continously eliminated, however, with NSW 4.0 showing a substantial improvement over 3.1.

3.2.4.1 MSG

MSG is a relatively stable MULTICS component. Its biggest problem is its dependence on the unsupported TASKING software. Unsupported items in the specification, as documented on October 3, 1978, do not appear to compromise the usability of the MULTICS TBH software - many remain unimplemented in other TBH systems.

Configuration control has been improved by creating a contact socket table so that MULTICS MSG can contact remote NSW MSG's at the correct socket numbers.

3.2.4.2 Foreman

The Foreman contains the greatest number of unimplemented items, and is the source of most problems on the MULTICS TBH. The implementation suffers from the fact that it was implemented to support tools written specifically for NSW - i.e. tools that use NSW tool primitives - and only later extended to support tool encapsulation. In general, encapsulation can now be done, but the quality of the encapsulation of each individual tool depends directly on the amount of work put into each encapsulation.

Specific improvements in the current implementation are:

- Many small bugs eliminated.
- Tool termination works essentially as specified.
- Alarm processing has been improved.
- More tools are encapsulated more reliably.

3.2.4.3 File Package

The File Package, like MSG, is a fairly reliable component. It conforms fairly closely to the specification, and supports file encodement into Intermediate Language about as well as the other TBH File Packages. Binary file transfer to non-MULTICS hosts is not supported, but is not required by any currently installed tools.

3.2.5 Front End

The COMPASS NSW Front End is not much different functionally then it was a year ago, since no major rewriting or addition of functions has been undertaken. It is, however, both faster and sturdier than it used to be:

Faster -- The FE program now handles (most of) its idle time by interrupt mechanisms rather than timed waits; hence it no longer consumes any CPU time between operations, and the CPU-time cost of waiting periods during operations has been cut in half.

Sturdier -- Anomalous conditions, especially in communications protocols, are detected more reliably and more discriminating responses are made. All known bugs have been corrected.

Several subtle accomodations have had to be made to the TOPS-20 operating system; but these have turned out to have no effect in the TENEX operating system, so that identical object-code files run on the two systems. Maintaining compatibility in this way means, of course, that no advantage has yet been taken of several of the advanced features offered by the newer TOPS-20.

Documentation:

The "external specs" of the FE are in reasonably good shape:

- The FE MSG Interface document, originally issued in November 1977, has been corrected and updated, and reissued in August 1978. It describes the format and content of all MSG messages sent or received by the FE. A further updating was issued in May 1979.

- The "user interface" document -- the NSW User's Reference Manual -- has been extended and partially rewritten, and was issued in November 1978 to describe the commands and operations available to the user in the NSW Version 3.1 release. This has been further updated and reissued as "NSW User's Reference Manual -- System Release 4.0" on 1 June 1979.

Shortcomings:

It is still possible for the FE process to "hang" if its conversational partner -- Works Manager or Foreman -- accepts an MSG message but then fails to reply. Without a moderately extensive rewriting of the programs, we are faced with the following choice in this circumstance:

(1) Abort the FE process, which leaves the user's Node Records in a blocked state so that he cannot log in again;

(2) Stop waiting for the reply and return to NSW command level: this works well for non-responsive Foreman during tool-termination scenarios, when the timeout has been set at 10 minutes; for Works Manager operations, however, this alternative leads to an out-of-synch situation from which the user cannot recover, if the belated reply does eventually arrive.

(3) Wait indefinitely for the reply, which is what we do now.

The program can still be made smaller and more efficient, and the input-editing facilities need to be completed.

3.3 NSW Performance

During 1978 a number of steps were taken to improve the overall performance of NSW. Three major avenues of approach were taken:

1. Memory use was monitored.

2. TENEX was monitored while running NSW in order to collect statistics on the gross use by NSW components of TENEX resources such as CPU time, JSYS monitor calls, and pager faults.

3. Detailed statistics were gathered on Works Manager CPU usage.

Memory use was monitored in two different ways. First, a memory monitoring tool called PAM was developed, and included in many NSW components. This tool, when activated, generates a map of exactly which virtual memory pages were accessed at least once between any two designated points in the execution of a program. This gives an accurate picture of the total number of memory pages that would be required to perform some NSW operation with no page faults. Because the result of using PAM is a map of exactly which pages were accessed, it is also possible to subdivide memory use into code and data accesses. From this it is possible to predict what the memory requirements would be for an NSW with a larger number of concurrent processes all of which shared code pages but each of which had its own local memory area.

PAM was able to show which pages were accessed at least once during an operation, but was unable to show how many times each page was accessed. Thus the figures obtained are doubtless larger than the true Working Set for NSW, in that pages are counted which may have been accessed only once or twice in an entire operation. In order to get a lower bound on NSW Working Set size, NSW was run on a metered version of TENEX and figures were obtained on the Working Set size that TENEX alloted to each NSW process. These figures represent a lower bound on the true Working Set, in that the figures also showed clearly that the TENEX configuration on which the tests were made had insufficient memory to run NSW without excessive paging. Unfortunately it is difficult to extrapolate from these figures just what the Working Set would be on a TENEX with adequate memory.

During 1978 BBN made a number of tests of overall system resource use by NSW. The results of these tests are described in great detail in

BBN Report No. 3847
A performance Investigation of the
National Software Works System
DRAFT VERSION
July 1978
Richard E. Schantz

In addition to the Working Set estimates already discussed, these tests showed that certain NSW processes were expending a great deal of time making JSYS calls to the operating system. As a result several NSW components, the File Package in particular, were altered to interact with the monitor more efficiently. This resulted in a substantial increase in File Package performance. These improvements are discussed in more detail in section 3.2.2.3 of this document.

These measurements of overall NSW component performance clearly showed that the Works Manager was consuming a large amount of CPU time, but gave no clue as to exactly where the time was being spent. To get a better picture of the problem a new performance tool for BCPL programs was developed: PFSTAT. PFSTAT takes samples of wall clock time, CPU time, and pager time at selected subroutine call and return points. The result is a detailed picture of what major subroutines were called and how much time each took to run. When PFSTAT was applied to the Works Manager it showed quite clearly that the major problem was that the Works Manager was using the powerful but slow Information Retrieval System to store all of its tables, including those tables which were accessed on every call. Accordingly, a new database management system called the Works Manager Table Facility was developed to hold the most active Works Manager tables, leaving the Information Retrieval System to handle only the NSW File Catalogue for which it was originally designed. As a result, the CPU time required by the Works Manager was reduced by a factor of 4.

4. Future Directions

4.1 Overview

As noted in section 2.3, we are now in phase five of NSW development: creation of a production NSW system. NSW needs to have the packaging, support, documentation, and capabilities of a finished production system. Phase five of NSW development will concentrate on providing these features. We began phase five in October, 1978 by beginning the expansion of NSW to support the activities of NSW implementors. The first specific improvements scheduled are the installation and testing of tools needed by the implementors, the addition of revision numbering of files, and extension of the NSW command language to support operations on groups of files. More details about specific features can be found in section 4.2.

In addition to program improvement, phase five includes the establishment of the administrative structure needed to support NSW users, manage the system configuration, operate systems, determine the priority of bug fixes and new features, prepare and distribute documentation, etc.

4.2 Components

In the following subsections we describe the tasks to be performed to complete phase four of NSW development and move into phase five.

4.2.1 Core System

4.2.1.1. Works Manager

Considerable effort was devoted to completion of phase four of Works Manager development. A number of measurements of Works Manager performance were made and analyzed. Substantial improvement was achieved upon completion of the in-core Works Manager Table Facility (see section 3.2.1.1). More performance optimization is possible, and more effort should be devoted to measurement, analyses, and implementations. Current efforts at modeling should also be continued.

In addition, certain portions of the full scale NSW reliability plan should be implemented. While portions of that plan treated distributed data base synchronization, other parts dealt with issues of process and network failure and recovery. These other parts should be implemented. In particular, the try-retry mechanism and timing signals are needed. Moreover, a facility for archiving and restoring NSW files and data bases should be designed and implemented.

Phase five, which is concerned with "productizing" NSW, began in October, 1978. While the Works Manager is substantially complete, there are a number of extensions which should be made. These enhanced capabilities include:

- Arpanet mail interface - The procedures to support mail systems (e.g., Hermes) should be designed and implemented.

- Configuration management procedures - As noted in section 3.1, manual configuration management has already begun. As more NSW development work is done using NSW, it will be possible to automate configuration management.

- Direct file access - Use access and read access: Add two new kinds of NSW file access. Use access means that a user has undisputed rights to an NSW file. When he references the file he is given the NSW file copy - not a private copy. Any alterations he makes are immediately reflected in the file. Read access allows a user to read the actual NSW file copy - not a private copy. Thus it is suitable for data base files.

- Tool kits - When a user runs a kit of several tools on one host, the workspace should be left unchanged between tools. Thus, intermediate files can be passed from tool to tool without delivery to NSW file space. Both of these features would greatly enhance and optimize the use of local tools.

- Revision numbers - Design and implement a file version numbering facility. This facility must be rich enough to support configuration management within NSW.

- History file - Implement the Works Manager routines to record information on the History File. Design and implement at least some interesting management/accounting routines which access this file.

- File groups - Extend the Works Manager command language so that whole groups of files can be copied, renamed, deleted, etc.

- Full file attributes - At present only the filename portion of the complete NSW filename can be used for retrieval. Also, the use of file attributes by tools is only permitted for the Global File Descriptor. The implementation of file attributes should be completed.

- Tool name extensions - The original concept of complete tool host transparency has proven unworkable. Thus, the notion of tool name should be extended to allow (explicit or implicit) host selection. By using the same mechanism as is used for files, the entire file lock system can also be used for tools.

- System status commands - The NSW user needs commands to interrogate system status and configuration: What tools are available? Which resources are up? What is the system load?

- Restructuring - At present, the Works Manager implementation does not support extended terminal dialog with the Front End. Communication of this sort needs to be simplified and optimized, especially regarding batch tool specification and execution of management tools.

- Optimization - Currently, the Works Manager accesses its File Catalog with TENEX/TOPS-20 Page Mapping operations. Alternatives, such as the privileged JSYS "DSKOP", should be evaluated. Some of the algorithms of the Information Retrieval System should be re-examined on the basis of actual NSW use.

- File space maintenance and management - Operations aids to file space maintenance (e.g. reconcilliation of the Works Manager's File Catalog with directories distributed at TBHs) and management (e.g. relocation of seldom used physical copies from hosts which are short of file space to those with surplus room) need to be implemented.

This list of WM extensions by no means exhausts the list of possible capabilities. Some of these extensions are scheduled f- implementation in 1979; other features will undoubtedly be suggested as NSW implementors begin to use NSW for their own development efforts.

4.2.1.2 Works Manager Operator

Very little needs to be done to complete phase four of Works Manager Operator development. The mechanism used for batch job submission has proven to be reliable in the face of Works Manager, network, and batch host failure. Various detail improvements are required, but these will not consume much effort. Moreover, performance of the Works Manager Operator has not been a problem, since it operates in background mode. The elapsed time for its operation is only a miniscule fraction of total batch job execution time. Some effort should be devoted to carefully measuring and reducing CPU utilization because of the possible effect on interactive NSW components, but this is not a high priority task. Documentation of the Works Manager Operator should be completed in the near future.

In phase five, it will be necessary to extend the functional capabilities of the Works manager Operator. Such extensions include:

- o Background file motion - The delays perceived by the user when files must be transferred or reformatted can be significantly reduced by performing such actions in background mode.
- o Job chaining - A desirable extension is to allow multiple batch tools to be run in sequence. Such a sequence should not be limited to just one batch host.
- o Device I/O - A variant of background file motion is to have WMO control input and output from devices local to a user.
- o Support of small (or non-NSW) batch hosts - Some hosts may be too small to support a Batch Job Package. Also, some hosts may be desirable as batch hosts but may not have the required NSW components (MSG, File Package). The Works Manager Operator should be extended to use existing Arpanet protocols (FTP, RJE) to submit batch jobs to such hosts.
- o File groups - Extend batch tool specification facilities so that groups of related files can be supplied to batch tools, using the same conventions described above (4.2.1.1).

4.2.2 TENEX/TOPS-20 TBH

4.2.2.1 MSG

Very little additional effort is required for TENEX/TOPS-20 MSG. There are still some outstanding MSG design issues:

- Details of MSG-MSG authentication - The general mechanism is as specified in the MSG design document of December, 1976. However, the details of the ARPANET protocol exchanges are being re-examined.

- Maximum message size - The maximum message size is specified to be 65536 bytes (2**16). No implementation will accept messages that large. At present there is informal agreement to limit message size to at most 2048 bytes.

- Process creation - This issue was skirted in the original specification. However, a satisfactory solution must be found which balances the dynamic cost of process initialization and the static cost of maintaining unused ready-to-run processes.

- Optimization techniques - Compound operations like "send then receive" should be added, and some MSG code could be included inside those processes run under MSG to reduce context switching.

- Reliability techniques - Allow for multiple hosts, process classes, or process instances to be considered as recipients of generically addressed messages (broadcasting), so that the system can function better in the presence of "downed" hosts. The NSW Fault Logger is an example of a process which could make good use of such a feature.

Once these design issues are resolved, TENEX/TOPS-20 MSG must be modified to incorporate them. In addition, recent performance measurements have suggested a number of improvements which should be implemented.

4.2.2.2 Foreman

Completion of phase four for the TENEX/TOPS-20 Foreman involves two tasks. The first is the integration of the reliability mechansims described in the full scale NSW reliability plan - in particular, the try-retry mechanism and timing signals. The second task is improving Foreman performance with respect to CPU utilization and paging requirements. A number of such improvements have been suggested by the measurements and analysis already done.

Although the TENEX/TOPS-20 Foreman substantially implements the specification, there are a number of additional capabilities which should be added. Some of these capabilities are implied by the specification, and some are additional. These capabilities include:

- Permanent integration of the TOPS-20 mountable structures interface
- Implementation of the solution to the saved LND workspace management problem
- Coordinated Works Manager/Foreman protocol design and implementation to have common data base items reflect local resource management decisions
- Implementation of tool-specific encapsulated tool interfaces to handle tool peculiarities and improve performance
- Direct tool interface to NSW functions - i.e., non-encapsulated tool interface
- Design and implementation of a Foreman modified for on-line tool debugging
- Design and implementation of Foreman extensions for tool kits.
- Incorporation of some of the file package's functionality in order to optimize file fetching and delivery operations.
- Exploit directory sub-groups on TOPS20 to optimize workspace allocation and escape from directory-imposed limits on the number of tools which can simultaneously run on TBH. Use a pseudo-archive to free space consumed by dead tools.

4.2.2.3 File Package

Functionally, the TENEX/TOPS-20 File Package is essentially complete, including implementation of IL encode/decode for binary files. Complete performance measurement and analysis must be done. Preliminary measurements have suggested some changes which should halve CPU utilization. Additional optimization should be performed. Some of the concepts of the reliability plan could also be extended to the File Package. The other major task to be completed in phase four is production of File Package documentation.

In phase five, the capability which should be added is direct output of files to the user terminal. Currently, an editing or display tool must be run; a Works Manager command like "TYPE <file-group>" should be added to allow transmission directly from File Package to Front End. Another task which should be pursued is optimization of cross-net file transfers. The baud rate of such transfers should be improved and automatic restart and backup procedures in case of file transmission errors should be designed and implemented.

4.2.3 IBM 360 TBH

All IBM 360 components need to be documented.

4.2.3.1 MSG

The IBM 360 MSG should have the deficiencies mentioned in section 3.2.3.1 repaired. In addition performance should be measured and improved. As the MSG design issues mentioned in section 4.2.2.1 are resolved, the IBM 360 MSG should be modified to reflect those resolutions.

4.2.3.2 Foreman

The IBM 360 Foreman implements only a small subset of the Foreman specification. To the extent that there is user interest in interactive tools on IBM 360 hosts, the Foreman should be extended to implement the entire specification. The Works Manager capabilities needed to support new interactive tools have been provided in NSW 4.0, and the installation of new tools is planned. These new tools will require file delivery (for the first time) and all necessary extensions to the 360 Foreman will be made accordingly.

4.2.3.3 File Package

The IBM 360 File Package is essentially complete. A few minor tasks remain to be done (see section 3.2.3.3), and these should be completed. Performance measurement, analysis, and improvement should be done. Output to terminal and optimization of cross-net file transfers should be done in conjunction with the TENEX/TOPS-20 File Package.

4.2.3.4 Batch Job Package

No further effort on this component seems necessary.

4.2.4 MULTICS TBH

As noted in section 3.2.4, the components of the MULTICS TBH have been baselined. It is now apparent that considerable effort must be devoted to making the Foreman implement the Foremen and Interim Reliability specifications. MSG and the File Package implementations are operating according to specification but, like other File Packages, the Multics File Package will have to implement the terminal output capability. All MULTICS components need to be documented.

4.2.5 Front End

Functionally, the TENEX/TOPS-20 Front End is essentially complete. It has also been completely instrumented. Measurements have been taken and analyzed. While some level of ad hoc performance improvement is possible, the current Front End, which started as only a debugging tool, must be completely restructured in order to obtain a satisfactory level of performance. The Front End is implemented as a multi-fork process. Almost all of these multiple forks can be collapsed into a single fork. This will decrease both CPU utilization and space requirements. Front End documentation should also be completed.

An additional path toward optimizing Front End performance is to split the Front End into the "switcher" and "parser" functions. A document describing the functionality of the split was produced in July, 1978. Since this split is orthogonal to the current fork structure, the reduction of the number of forks should be completed before considering the implementation of the split Front End.

Parts of the full scale NSW reliability plan also must be implemented in the TENEX/TOPS-20 Front End - in particular, the try/retry mechanism and timing signals. With the completion of these performance and reliability tasks, phase four of Front End development will be finished.

There are several Front End enhancements which should be accomplished as part of phase five of NSW development. These enhancements include:

- Optimization of local tool use - Some advantage should be taken when the Front End and task are on the same host. The split Front End is an approach to this optimization.

- Macro facility - An NSW macro facility should be designed and implemented. This would permit users to execute a number of system/tool commands with a single command. It should be able to execute either online or in background mode.

- User profiles - Use of the user profile to tailor terminal handling should be designed and implemented.

- Access to text files - Currently the Front End can't access NSW files - if the user wishes a file listed, an editor or display tool must be invoked. The Front End should be able to list the file itself, and additionally should be able to take commands from a file to implement the "Runfile" capability discussed later (see 4.3.3).

- Status character - A control character should be available at all times to report on status. Certainly connection state can be supplied (Front End types out "FE alive") and possibly a report on Front End MSG status.

- Control character handling - The Works Manager uses control characters for certain functions (abort, test status, get attention, etc). The native systems which serve as TBHs use conventions which resemble the Works Manager's but often with local variation. Some tools have private uses for control characters. A novice user would like standardization, but an expert will want to use full keyboard functionality. This issue needs study.

4.3 Integration Testing

4.3.1. History

COMPASS has been responsible since mid 1977 for Integration Testing of NSW as outlined in "National Software Works Test Plan", May 9, 1977, published by RADC/ISCP. Since that date, COMPASS has run a manual Integration Testing script on each version of the NSW system which was a candidate for release as a new user system.

The initial version of this script was restricted to the level of test specified in RADC/ISCP Test Plan - to determine if NSW components functioned as specified in a friendly environment. Testing was limited to ensuring that all components in the test configuration (including remote TBH's) responded correctly to correct user input, and little effort was made to test the system in the face of incorrect input or errors in the system configuration. NSW systems tested to only this level tended to behave erratically. Therefore the Integration Test script was soon extended with a number of ad hoc tests of NSW's capacity to cope with user and configuration errors. This is the level of testing to which the candidate user system released on November 16, 1978 was subjected.

COMPASS has been mandated to develop and apply a more carefully designed and rigorous level of Integration Testing to future NSW system releases. The remainder of this section describes the direction for this Integration Testing.

4.3.2. Functional tests - content

We define "integration testing" as follows: to determine whether a set of NSW components offered as a new system release meet the following requirements:

(1) Can be correctly configured as an operational NSW system, with all core and TBH components in a correct initial state for operation.

(2) All functions specified to be present in the release perform as expected for correct input, and all components in the configuration function as specified for correct input.

(3) All error detection and reporting functions work as expected for representative incorrect (user) input. All components report and recover from user induced errors as specified.

(4) The interim reliability scenarios perform as specified.

'5) The system recovers from configuration failures (e.g. TBH crashes) to the extent specified and expected for the release.

This testing includes complete tests for the delivery system for tools at each TBH - Foreman, File Package, Batch Job Package, etc - but does not cover acceptance of any tools.

The test scripts are structured into a series of levels; the first level tests the least functionaity and the least complex core of the configuration. Each succeeding level tests more functionality and/or more of the system configuration. The folowing scripts were used to test the candidate user system released to PDC on May 1, 1979.

The general contents of the scripts is as follows:

Level 0: Set up the complete system configuration, and verify that all components are in a proper initial executive and communications state.

Level 1: Test core system: all components local on Works Works Manager host.

(a) Test all possible NSW command paths with correct input in the following order:

i. LOGIN, MOVE, CHANGE password, LOGOUT.

ii. Project management tools: nodes, assign rights, etc.

iii. ALTER command - SCOPE manipulation.

iv. File commands - NET, RENAME, COPY, DELETE, SEMAPHORE. Local file transfe's only.

v. Enter a batch job. (Processing deferred).

vi. Use a local interactive tool. Test slewing, multiple tools, RESUME.

All recognition and completion features of the Front End are to be tested.

(b) Recapitulate relevant sections of (a), with representative errors on input. The error detection and reporting facilities of the local components are to be tested in the following order:

i. Front End
ii. Works Manager
iii. File Package
iv. Foreman

(c) Where appropriate in (a) and (b), the operation of the Checkpointer is to be monitored, and message and error logging is to be monitored.

Level 2: Test the distributed system: at least one instance of each TBH family to be involved in the configuration.

(a) File transfer tests

i. Test family transfers, where available. Currently limited to TENEX/TOPS20 hosts due to lack of multiple host resources.

ii. Test non-family transfers. At least one text file transfer back and forth between each family pair in configuration, and one round-robin transfer in a chain including all families. Multiply translated files must be identical under Intermediate Language semantic specification. At least one binary file transfer of each defined type.

(b) TBH test

i. Execute a batch job at each BTBH. Monitor performance of Works Manager Operator and Batch Job processor for each job.

ii. Execute one interactive tool at each TBH. Level of test identical to 1 (a) vi.

(c) Recapitulate (a) and (b) introducing representative errors in user input.

Level 3: Test interim reliability scenarios. Induce each error condition covered by interim reliability plan, and monitor all components involved for correct behavior.

(a) Initial test will be for the core system only, particularly to test correct behavior of Works Manager.

(b) Test of Foreman capability for each TBH. Induce only those failures which test the Foreman's role in the reliability scenarios.

Level 4: Test system response to induced configuration failures. Beyond checking response to "crashed" TBH (NSW taken down), the content of this test level is to be specified.

4.3.3. Functional tests - methodology

It will be necessary to automate these tests as much as possible both to avoid expending excessive professional staff time on them, and to make the tests reliably repeatable. COMPASS has investigated three classes of tools which can assist this automation effort:

1. Run file facilities external to NSW:

   TENEX RUNFIL
   TOPS20 TAKE
   TELNET take.input.from.file

2. Run file facilities within NSW.

   Front End RUNFILE command

3. Production (syntactic rule) systems

   RITA

1. Run file facilities external to NSW

The tools listed are all basically similar. Each has the advantage of being familiar, tested and straightforward. All lack a sufficiently sophisticated means of synchronizing their input to the processes they control with what is in fact happening. The synchrony problem limits these tools to situations in which no slewing between TELNET connections is done. This excludes any testing of NSW tools, and makes changing TELNET conversational partners to monitor configuration status changing unreliable.

2. Run file facilities within NSW

Provision of a RUNFILE command has one outstanding advantage: the Front End is always aware of the identity of the user's conversational partner-NSW command processor, HELP call, or tool - and is thus perfectly placed to control the synchronization of command file with the actual behavior of NSW. An additional advantage is that we can add desired features to this facility as needed, but must accept the others as they are. The disadvantages are that this facility has to be designed, implemented and tested; and that it can only automate the user input portions of the test scripts.

3. Production systems - RITA

RITA has the advantage that it can handle both user input and configuration management with a sufficiently rich rule set. Our studies indicate that the development of such a rule set would be a demanding job. A more significant problem is that TENEX RITA is likely to consume excessive CPU recources to run a rule set as complex as that needed by NSW.

Proposed Methodology

We propose that a mixture of manual testing and the use of two of the tools described above be used to run the functional tests. The mix would be as follows:

1. Use RITA to set up and initialize the NSW configuration for each level test, and confirm that the initialization is correct.

2. Use NSW RUNFILE to automate all user input to test Levels 1, 2, and 3. The RUNFILE facility will have some or all of the following features:

   (i). Ability to interrupt

   (ii). A synchronization scheme

   (iii). HELP from attached user if synchronization failure occurs

   (iv). A PAUSE feature

   (v). A macro feature - string and/or file name binding at run time.

   (vi). A "learning" feature which will allow the Front End to do most of the work of turning a manual script into a command file (speculative).

3. Use manual scripts for much of level 3 testing and most of level 4 testing. Probe system status and monitor component operation as required during Level 1, 2, and 3 testing.

4.4 Miscellaneous

There are additional tasks to be undertaken which do not fall within the scope of a single component. One major effort, the creation of an administrative structure for NSW, was mentioned in section 4.1. In this section we list some additional efforts:

- Help facility - an online help mechanism for NSW users should be designed and implemented. This should probably look like a tool within NSW.

- Distributed system debugger - It should be possible to debug a distributed system like NSW from within NSW. An appropriate debugger should be designed and implemented. This will almost certainly require changes to the Works Manager and Foreman components, and possibly to MSG also.

- Automated testing - The functional and stress/regression testing of NSW test and user systems should be automated.

- Management tools - Tools for manipulating the project tree are available in rudimentary form. These should be improved, and additional tools for accessing the History file, report generation, etc. designed and implemented.

- Operators' tools - A tool kit for the user system operator to at least partially automate data base cleanup, system starting, etc. should be designed and implemented.

- Tool installation - Install, test, and document more NSW tools. In particular, install a tool kit adequate for NSW implementors.

Bibliography

(1) Shapiro, R.M., and Millstein, R.E., "Plan for Hardening, Scaling and Optimizing the Works Manager", Massachusetts Computer Associates, Inc., CADD-7601-1511, January 15, 1976.

(2) NSW Protocol Committee, "MSG: The Interprocess Communications Facility for the National Software Works (Preliminary)", Massachusetts Computer Associates, Inc., CADD-7601-2611, and Bolt, Beranek and Newman, Inc., 3237, January 23, 1976.

(3) Warshall, S, "NSW: Tools for Management Support", Massachusetts Computer Associates, Inc., CADD-7602-0211, February 2, 1976.

(4) Muntz, C.A., and Cashman, P.M., "File Package: The File Handling Facility for the National Software Works (Preliminary)", Massachusetts Computer Associates, Inc., CADD-7602-2011, February 20, 1976.

(5) "Semi-Annual Technical Report", Massachusetts Computer Associates Inc., CADD-7603-0411, March 4, 1976.

(6) Schantz, R.E., and Millstein, R.E., "The Foreman: Providing the Program Execution Environment for the National Software Works (Preliminary)", Bolt, Beranek and Newman, Inc., 3266, and Massachusetts Computer Associates, Inc., CADD-7604-0111, March 31, 1976.

(7) Warshall, S., "NSW: Tools for Management Support (1st revision)", Massachusetts Computer Associates, Inc., CADD-7604-0811, April 8, 1976.

(8) Millstein, R.E., and Schantz, R.E., "NSW Tool Builder's Guide (Preliminary)", Massachusetts Computer Associates, Inc., CADD-7605-2111, and Bolt, Beranek and Newman, Inc., 3308, May 21, 1976.

(9) Geller, D.P., "NSW User's Guide (Preliminary)", Massachusetts Computer Associates, Inc., CADD-7605-3111, May 31, 1976.

(10) Geller, D.P., "NSW Manager's Guide (Preliminary)", Massachusetts Computer Associates, Inc., CADD-7605-3112, May 31, 1976.

Geller, D.P., "Preliminary NSW User's Guide", Massachusetts Computer Associates, Inc., CADD-7606-1111, June 11, 1976.

Geller, D.P., "Preliminary NSW Manager's Guide", Massachusetts Computer Associates, Inc., CADD-7606-1112, June 11, 1976.

(11) Warshall, S., "NSW: Tools for Management Support (A Final Report)", Massachusetts Computer Associates, Inc., CADD-7607-0111, July 1, 1976.

Millstein, R.E., "The National Software Works", Massachusetts Computer Associates, Inc., CA-7607-3011, July 30, 1976.

(12) "Second Semi-Annual Report", Massachusetts Computer Associates, Inc., CADD-7608-1611, August 16, 1976.

Geller, D.P., "Interim NSW User System", Massachusetts Computer Associates, Inc., CA-7609-2911, September 29, 1976.

Geller, D.P., "Interim NSW Managers Tools", Massachusetts Computer Associates, Inc., CA-7609-2912, September 29, 1976.

(13) Sattley, K., "NSW Prototype Management Tools - Subsystem Specification", Massachusetts Computer Associates, Inc., CADD-7610-0711, October 7, 1976.

(14) Sattley, K., "NSW Prototype Management Tools - Maintenance Manual", Massachusetts Computer Associates, Inc. CADD-7611-0411, November 4, 1976.

(15) NSW Protocol Committee, "MSG: The Interprocess Communication Facility for the National Software Works (1st revision)", Massachusetts Computer Associates, Inc., CADD-7612-2411, and Bolt, Beranek and Newman, Inc., 3237, December 24, 1976.

(16) Cashman, P.M., and Faneuf, R.A., and Muntz, C.A., "File Package: The File Handling Facility for the National Software Works (1st revision)", Massachusetts Computer Associates, Inc., CADD-7612-2711, December 27, 1976.

(17) Schantz, R.E., and Millstein, R.E., "The Foreman: Providing the Program Execution Environment for the National Software Works (1st revision)", Bolt, Beranek and Newman, Inc., 3442, and Massachusetts Computer Associates, Inc., CADD-7701-0111, January 1, 1977.

Shapiro, R.M. and Millstein R.E., "NSW Reliability Plan", Massachusetts Computer Associates, Inc., CA-7701-1411, January 14, 1977.

Millstein, R.E., and Shapiro, R.M., "Interim NSW Reliability Plan", Massachusetts Computer Associates, Inc., CA-7701-2111, January 21, 1977.

Bearisto, D.B., "The Front End: User's Interface to the National Software Works", Massachusetts Computer Associates, Inc., CA-7701-2112, January 21, 1977.

Millstein, R.E., and Shapiro, R.M., "Interim NSW Reliability Plan (1st revision)", Massachusetts Computer Associates, Inc., CA-7701-2111, February 8, 1977.

(18) Millstein, R.E., and Schantz, R.E., "NSW Tool Builder's Guide (1st revision)", Massachusetts Computer Associates, Inc., CADD-7702-1811, and Bolt, Beranek and Newman, Inc., 3308, February 18, 1977.

(19) "Third Semi-Annual Technical Report for the National Software Works", Massachusetts Computer Associates, Inc., CADD-7702-2811, February 28, 1977.

(20) Geller, D.P., "Preliminary NSW User's Guide Volume-II: NSW Language Reference Manual", Massachusetts Computer Associates, Inc., CADD-7612-3011, December 30, 1977.

Millstein, R.E., and Shapiro, R.M., "Interim NSW Reliability Plan (2nd revision)", Massachusetts Computer Associates, Inc., CA-7701-2111, March 1, 1977.

Shapiro, R.M., and Millstein, R.E., "NSW Reliability Plan (1st revision)", Massachusetts Computer Associates, Inc., CA-7701-1411, June 10, 1977.

(21) Geller, D.P., "NSW Functional Test Plan/Procedures", Massachusetts Computer Associates, Inc., CADD-7706-3011, June 30, 1977.

Millstein, R.E., "The National Software Works: A Distributed Processing System", Massachusetts Computer Associates, Inc., CA-7708-0911, August 9, 1977.

Faneuf, R.A., "The National Software Works: Operational Issues in a Distributed Processing System", Massachusetts Computer Associates, Inc., CA-7708-1911, August 19, 1977.

Geller, D.P., "The National Software Works", Massachusetts Computer Associates, c., CA-7708-1111, August 11, 1977.

Shapiro, R.M., and Millstein, R.E., "Reliability and Fault Recovery in Distributed Processing", Massachusetts Computer Associates, Inc., CA-7708-1811, August 18, 1977.

Schaffner, S., "BCPL Standards for NSW", Massachusetts Computer Associates, Inc., CAID-7708-2311, August 23, 1977.

(22) Guerrieri, M., Schaffner, S., and Sluizer, S., "Works Manager Program Maintenance Manual", Massachusetts Computer Associates, Inc., CADD-7709-2711, September 27, 1977.

(23) Guerrieri, M., Schaffner, S., and Sluizer, S., "Works Manager Subsystem Specifiction", Massachusetts Computer Associates, Inc., CADD-7709-2712, September 27, 1977.

(24) Schaffner, S., "Works Manager Database Specification", Massachusetts Computer Associates, Inc., CADD-7709-2713, September 27, 1977.

(25) Bearisto, D., "System/Subsystem Specification; WMO & IBS Methodology; WMO & IBS Documentation Program Maintenance Manual; WMO & IBS System Documentation", Massachusetts Computer Associates, Inc., CADD-7709-2714, September 27, 1977.

(25a) Geller, D.P., and Sattley, K., "NSW User's Reference Manual System Version 2.1", Massachusetts Computer Associates, Inc., CADD-7710-2611, October 26, 1977.

Shapiro, R.M., and Millstein, R.E., "Failure Recovery In A Distributed Data Base System", Massachusetts Computer Associates, Inc., CA-7712-0911, December 9, 1977.

Schaffner, S., Sluizer, S., and Guerrieri, M., "NSw Utilities Program Maintenance Manual", Massachusetts Computer Associates, Inc., CADD-7804-0111, April 1, 1978.

Sluizer, S., Guerrieri, M., and Schaffner, S., "Information Retrieval System Program Maintenance Manual", Massachusetts Computer Associates, Inc., CADD-7804-0112, April 1, 1978.

(26) Geller, D.P., and Sattley, K., "NSW User's Guide - Addenda", Massachusetts Computer Associates, Inc., [This document was added to USERGUIDE-2.1-doc on April 7, 1978. With the inclusion of these addenda, that document becomes the NSW User's Reference Manual for System Version 2.4.]

(27) Shapiro, R.M., and Millstein, R. E., "NSW Design Plan", Massachusetts Computer Associates, Inc., CADD-7804-1011, CADD-7804-1011, April 11, 1978.

(28) "Interim Report for the National Software Works", Massachusetts Computer Associates, Inc., CADD-7805-0111, May 1, 1978.

(29) Cashman, P., "An Analysis of the NSW System Release Procedure", Massachusetts Computer Associates, Inc., CA-7810-2511, October 25, 1978.

(30) Schaffner, S., "Works Manager Subsystem Specification", Massachusetts Computer Associates, Inc., CADD-7709-2712, September 27, 1977, revised November 15, 1978.

(31) Geller D., Sattley, K., "NSW Users Reference Manual System Version 3.1", Massachusetts Computer Associates Inc., CADD-7811-1411, November 14, 1978.

(32) Sluizer, S., Guerrieri, M., Schaffner, S., "Information Retrieval System Program Maintenance Manual", Massachusetts Computer Associates, Inc., CADD-7804-0112, April 1, 1978, revised November 16, 1978.

(33) Schaffner, S., Sluizer, S., Guerrieri, M., "NSW Utilities Program Maintenance Manual", Massachusetts Computer Associates, Inc., CADD-7804-0111, April 1, 1978, revised November 16, 1978.

(34) Guerrieri, M., Schaffner, S., Sluizer, S., "Works Manager Program Maintenance Manual", Massachusetts Computer Associates, Inc., CADD-7709-2711, September 27, 1977, revised November 16, 1978.

(35) Cashman, P., Sluizer, S., Myszewski, M., "Intran Language Manual", Massachusetts Computer Associates, Inc., CAID-7812-0111, December 1, 1978.

(36) Cashman, P., Sluizer, S., "The Table Definition Language Manual", Massachusetts Computer Associates, Inc., CAID-7812-0111, December 1, 1978.

(37) Cashman, P., Sluzier, S., "Intran Inserter User's Guide", Massachusetts Computer Associates, Inc., CAID-7812-0113, December 1, 1978.

(38) Cashman, P., Sluizer, S., "TDL User's Guide", Massachusetts Computer Associates, Inc., CAID-7812-0114, December 1, 1978.

(39) Muntz, Charles, A., "Final Report on the National Software Works", for the period October 1, 1977 - December 6, 1978, Massachusetts Computer Associates, Inc., CADD-7901-0511, January 5, 1979.

(40) Cashman, Paul. M., "A Review of the Literature on the Software Quality Assurance and Related Topics, Massachusetts Computer Associates, Inc., CA-7902-1611, February 16, 1979, revised March 29, 1979.

(41) Cashman, Paul. M., "An Analysis of the NSW Configuration Management Plan", Massachusetts Computer Associates Inc., CA-7902-2611, February 26, 1979.

(42) Cashman, Paul. M., "A Tutorial on the Military Standards for Software Acquisition", Massachusetts Computer Associates, Inc., CA-7904-1111, April 11, 1979.

(43) Tsao, Nina, "Checkpointer Subsystem Specification", Massachusetts Computer Associates, Inc., CAID-7904-1611, April 16, 1979.

(44) Faneuf, Ross, "Release Specific Documentation: NSW Release 4.0", Massachusetts Computer Associates, Inc., CA-7905-0411, May 4, 1979.

(45) Schaffner, Stuart, "FSDPKG: Support for NSW Filespace Designators", Massachusetts Computer Associates, Inc., CADD-7906-0111, June 1, 1979.

(46) Schaffner, Stuart, "The Information Retrieval System: Support for the NSW File Catalogue Database Program Maintenance Manual", Massachusetts Computer Associates, Inc., CADD-7906-0112, June 1, 1979.

(47) Schaffner, Stuart, "Works Manager Table Facility Design Version 2.0", Massachusetts Computer Associates, Inc., CADD-7906-0113, June 1, 1979.

(48) Schaffner, S., Tsao, N., and Faneuf, R., "NSUPKG: The NSW Utilities Package for BCPL Programs", Massachusetts Computer Associates, Inc., CADD-7906-0114, June 1, 1979.

(49) Schaffner, Stuart, "BCPPKG: Utilities for BCPL Programs", Massachusetts Computer Associates, Inc., CADD-7906-0115, June 1, 1979.

(50) Schaffner, Stuart, "WMTPKG: Support for Works Manager Table Access", Massachusetts Computer Associates, Inc., CADD-7906-0116, June 1, 1979.

(51) Schaffner, S., and Sluizer, S., "Works Manager Subsystem Specification", Massachusetts Computer Associates, Inc. CADD-7906-0117, June 1, 1979.

(52) Sluizer, Suzanne, "The Works Manager Procedures: Externally Callable Routines in the Works Manager Program Maintenance Manual", Massachusetts Computer Associates, Inc., CADD-7906-0118, June 1, 1979.

(53) Tsao, Nina, "Checkpointer Subsytem Specification", Massachusetts Computer Associates, Inc., CADD-7906-0119, June 1, 1979.

(54) Sattley, K., and Geller, D.P., "NSW User's Reference Manual System Version 4.0", Massachusetts Computer Associates, Inc., CADD-7906-0120, June 1, 1979.

(55) Sluizer, S., "Works Manager Interface Conformance Report", Massachusetts Computer Associates, Inc., CA-7906-0111, June 1, 1979.

(56) Tsao, Nina, "Checkpointer Interface Conformance Report", Massachusetts Computer Associates, Inc., CA-7906-0112, June 1, 1979.

(57) Faneuf, Ross, "File Package Interface Conformance Report", Massachusetts Computer Associates, Inc., CA-7906-0113, June 1, 1979.

(58) Sattley, Kirk, "Front End Interface Conformance Report", Massachusetts Computer Associates, Inc., CA-7906-0114, June 1, 1979.

(59) Cashman, Paul M., "MONSTR: The NSW Tool to Monitor Software Trouble Reporting: A-level System Specification", Massachusetts Computer Associates, Inc., CA-7907-0911, July 9, 1979.

(60) "Semi Annual Technical Report on the National Software Works for the period 12 December 1978 through 30 June 1979", Massachusetts Computer Associates, Inc., CADD-7907-1201, July 12, 1979.

Glossary

Item	Section where defined
Architecture Control Contractor (ACC)	2.3.5
Batch Job Package (BJP)	2.2.1
Checkpointer (CHKPTR)	3.2.1.2
Configuration Database	3.2.1.6
Development and Maintenance Contractor (DMC)	2.3.5
Fault Logger (FL)	3.2.1.5
File Package (FP)	2.2.1
Foreman (FM)	2.2.1
Front End (FE)	2.2.1
Immersion	2.1.2
Interactive Batch Specifier (IBS)	2.3.2
Interface Protocol (IP)	2.3.2
MSG	2.2.1
National Software Works (NSW)	1.
NSW Operations (NSWOPS)	2.3.5
NSW Standard Transaction (NST)	2.3.5
Policy Group (PG)	2.3.5
Product Development Contractor (PDC)	2.3.5
Release Specific Document	2.3.5
Software Trouble Report (STR)	2.3.5
Tool Bearing Host (TBH)	2.3.2
Tool Manager (TM)	2.3.5
Works Manager (WM)	2.2.2
Works Manager Operator (WMO)	2.3.2