

U.S. DEPARTMENT OF COMMERCE
National Technical Information Service

AD-A033 085

DISTRIBUTED COMPUTATION AND TENEX-RELATED
ACTIVITIES

BOLT BERANEK AND NEWMAN, INCORPORATED
CAMBRIDGE, MASSACHUSETTS

AUGUST 1976

349061

BCLT BERANEX AND NEWMAN INC

CONSULTING · DEVELOPMENT · RESEARCH

ADA U33085

BBN Report No. 3450

August 1976

DISTRIBUTED COMPUTATION AND TENEX-RELATED ACTIVITIES

Quarterly Progress Report No. 7

1 May 1976 to 31 July 1976

DDDC
RECEIVED
DEC 9 1976
RECEIVED
A

The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied of the Defense Advanced Research Projects Agency or the United States Government.

This research was supported by the Defense Advanced Research Projects Agency under ARPA Order No. 2901, Contract No. N00014-75-C-0773.

Distribution of this document is unlimited. It may be released to the Clearinghouse, Department of Commerce for sale to the general public.

REPRODUCED BY
NATIONAL TECHNICAL
INFORMATION SERVICE
U. S. DEPARTMENT OF COMMERCE
SPRINGFIELD, VA. 22161

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER BBN Report No. 3450	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) DISTRIBUTED COMPUTATION AND TENEX-RELATED ACTIVITIES		5. TYPE OF REPORT & PERIOD COVERED Quarterly Progress 5/1/76 - 7/31/76
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) R. Schantz, R. Thomas	8. CONTRACT OR GRANT NUMBER(s) N00014-75-C-0773	
9. PERFORMING ORGANIZATION NAME AND ADDRESS Bolt Beranek and Newman Inc... 50 Moulton Street Cambridge, Massachusetts 02138	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS	
11. CONTROLLING OFFICE NAME AND ADDRESS	12. REPORT DATE August 1976	
	13. NUMBER OF PAGES 14	
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)	15. SECURITY CLASS. (of this report) Unclassified	
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE	
16. DISTRIBUTION STATEMENT (of this Report) Distribution of this document is unlimited. It may be released to the Clearinghouse, Department of Commerce for sale to the general public.		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES This research was supported by the Defense Advanced Research Projects Agency under ARPA Order No. 2935.		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) distributed computation distributed operating system National Software Works TENEX operating system		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) This report describes BBN efforts in the design of the National Software Works system and BBN efforts to integrate TENEX into the National Software Works system.		

TABLE OF CONTENTS

	<u>Page</u>
1. Introduction.	1
2. MSG: The NSW Interprocess Communication Facility	4
3. The Foreman: Supporting NSW Tool Execution	10
4. Other Activities.	14

ACCESSION for	
NTIS	NTIS Section
DOC	DOC Section
UNANNOUNCED	
JUSTIFICATION	
BY	
DISTRIBUTION PRIORITY CODES	
DATE	APPROVED BY
<i>A</i>	

1. Introduction

Our participation in the National Software Works (NSW) project is divided into three major areas. These are system design, system implementation, and contractor/sponsor meetings coordinate design and implementation goals and strategies. During this quarter most of our effort has been in the latter two areas.

The primary focus for our work this quarter was preparation of an NSW system for demonstration to ARPA and Air Force personnel in July. The NSW system demonstrated in July was a significant milestone because it represented the first multiple computer NSW. Our preparations for the July demonstration included enhancements to the TENEX implementation of MSG [the NSW inter-host communication facility (See BBN Reports No. 3237 and No. 3315)] and completion of the initial implementation of a TENEX NSW Foreman for the multi-computer NSW environment (See BBN Reports No. 3260 and No. 3237). The work on TENEX MSG and the TENEX Foreman is described in detail in Sections 2 and 3 of this report.

During this quarter we consulted with personnel from UCLA and MIT as they worked on initial implementations of MSG for the 360/91 and Multics hosts. We assisted them by resolving ambiguities in the MSG design specification, by suggesting

implementation strategies based on our experience with the TENEX MSG implementation, and by providing access to the TENEX MSG for exercising their implementations. In addition, there were NSW technical briefings during this quarter to personnel from both Rome Air Development Center and the ARPA office.

In an NSW related area, we began to consider the problem of transferring the software we have been developing to make TENEX an NSW tool-bearing host to the new DEC TOPS-20 operating system. TOPS-20 is the operating system for a new line of DEC hardware designed to replace (and be upward compatible with) the KA10 and KI10 processors. It evolved from an early version of TENEX (approximately 1972).

The goal here is to take advantage of the TENEX NSW effort to make TOPS-20 a tool-bearing host with a minimal additional investment in software development. Ideally, we would like to be able to run the TENEX NSW software, with no modification, under TOPS-20. The problem here is that the TENEX NSW software makes use of features added to TENEX after DEC began to develop TOPS-20 and which, as a result, are not presently supported by the TOPS-20 system. As a first step in transferring the TENEX NSW software to TOPS-20, we have compiled a list of those TENEX features that are missing from TOPS-20 which are used by the NSW software. We have submitted this list both to ARPA and to DEC. This list represents a starting point for discussing possible enhancements to the TOPS-20 system.

Further cooperation between BBN, DEC, and ARPA will be required to determine in detail how TOPS-20 will be modified and how the TENEX NSW software will be modified to accomplish the transfer. We are optimistic that the NSW software will be transferred to TOPS-20 in a cost effective way.

2. MSG: The NSW Interprocess Communication Facility.

During the previous quarter we completed an initial implementation of MSG for TENEX. This was described in our last progress report (See BBN Report No. 3315). That initial implementation was not a full implementation of all the features specified in the MSG design document. The features implemented were chosen to support preliminary debugging and integration of inter-host versions of the major NSW system components. The TENEX MSG implementation has been improved considerably in two areas during the last quarter: a number of functions not present in the initial MSG implementation have been added; a number of features to facilitate the use of MSG and the debugging of NSW processes have been added.

The ability to support "direct connections", as specified by the MSG design document, has been added to the TENEX MSG. Direct connections between processes provide an efficient means for the connected processes to engage in a long term conversation. After such a connection is established the processes can communicate directly with no further assistance from MSG. This is in contrast to message communication which requires MSG action for each transmitted message. These direct connections are supported by ARPANET host/host protocol connections. MSG provides processes a means to establish direct connections which makes the

details of host/host protocol (such as socket numbers, RFCs, etc.) transparent to the processes. To establish a connection a process need only specify to MSG the name for the target process and the connection type desired (i.e., user TELNET, server TELNET, binary send or receive, binary send/receive pair, etc.). MSG itself then acts to establish the requested connection and to notify the process when the connection is ready for use.

The NSW File Package processes use this feature to establish direct connections which are used for inter-host file transfers. In addition, communication between users (at Front Ends) and interactive tools will be supported by MSG TELNET connections. The Front End and tool/Foreman processes currently use an ad hoc protocol to establish the user-to-tool communication path. These processes will be simplified shortly to use the MSG direct connection facilities.

The MSG "Rescind" primitive was added to the TENEX implementation. This primitive allows a process to abort a previously requested action such as an attempt to send or receive a message. Execution of Rescind instructs MSG to delete the "pending event" associated with the action in question and allows MSG to recover the system resources allocated to the request. This primitive will be primarily used by processes in failure recovery procedures.

We have implemented an interface between the NSW Dispatcher and MSG. Recall that to access NSW a user (via a process acting on his behalf, such as TELNET or a TIP) interacts with the NSW Dispatcher. The Dispatcher creates a job for a Front End process and connects the user to it. The Front End, of course, runs under the control of MSG.

As described in our last progress report, an instance of MSG and the NSW processes it supports are implemented on TENEX by a number of cooperating jobs. One of these, the "central" MSG job (refer to BBN Report 3315 for details), is created automatically when TENEX is restarted (e.g., after a scheduled shutdown or a crash). The central MSG job creates the other process managing MSG jobs as part of its initialization procedure. The Dispatcher-MSG interface provides the means by which new MSG jobs are dynamically created to run Front End processes for users.

A goal of the NSW system implementation is to be resilient to the failure of system components. One aspect of resilient behavior is continued operation in the presence of host system crashes. Another important aspect is the integration of crashed hosts back into the system after they have been restarted. To achieve this re-integration, NSW components resident on a host that has been restarted must perform special crash recovery procedures before they can resume their normal functions. For example, after its host restarts, the host Foreman should attempt to save user files that may have been trapped in tool workspaces when the host crashed.

For a component to be able to execute its recovery procedure, it must be notified that its host has just restarted. A mechanism has been added to MSG to do this notification. When configuring an NSW system, the generic process names known to the system (e.g., Works Manager, Front End, etc.) must be declared. The declaration of a particular generic process class may specify that a process in the class should be started automatically whenever MSG restarts. (As far as NSW is concerned, an MSG restart is equivalent to a host restart.) If this specification is made, MSG will, upon completion of its own initialization, create an instance of the process. It does this by simulating a generically addressed message for the process class. The process created to receive the message can detect that it is the "first" process in its class, and that it should execute its recovery procedure, by noting that the message is specially marked as from MSG (rather than from some NSW process requesting service).

A process monitoring and debugging facility has been added to MSG. The user (i.e., an implementer of an NSW component such as the Works Manager or the Foreman) can access this facility by typing a control character (CNTL-S) which activates a command language interpreter for MSG. Using this command language interpreter the user can query the status of the various jobs which compose the MSG configuration (see BBN Report No. 3315 for MSG implementation details) and the status of the processes currently managed by MSG. The process status information reported includes the process name, the process state (program

counter plus execution status), and any "pending events" created as the result of previously executed, but not yet completed, MSG primitives (such as send and receive operations). In addition, a user can request a listing of the primitives recently executed by MSG processes. This list is printed in reverse chronological order (most recent first). For each primitive, it includes the particular primitive (e.g., ReceiveGeneric, SendSpecific), the name of the executing process, the name of the target process (if appropriate), the time the primitive was executed, the outcome of the primitive (i.e., success or failure plus reason), and other relevant information.

Although at present this facility provides only capabilities for monitoring process activity, MSG users have found it to be very useful in debugging their processes. During the next quarter we plan to increase the utility of the facility by adding capabilities to allow users to exercise control over (e.g., start, stop, kill) as well as to invoke debuggers (such as DDT, IDDT, BDDT) for individual processes.

In order to facilitate the checkout of new versions of MSG we have developed a set of MSG diagnostic programs. These programs execute as a pair of communicating MSG processes which exercise most of the MSG features. The algorithms for these processes have been documented so that other MSG implementers can build versions of them. Versions of the diagnostics for the 360/91 and Multics hosts have proven quite useful in debugging

the MSG implementations for those machines. We recommend that each MSG implementation include an implementation of these diagnostics.

Finally, we have prepared and distributed preliminary user documentation for the TENEX MSG implementation. We plan to release more comprehensive MSG user documentation at the end of the calendar year.

3. The Foreman: Supporting NSW Tool Execution

During this reporting period the TENEX Foreman for the multi-host NSW environment became operational. The TENEX Foreman is, at present, a partial implementation in the sense that it does only a minimum with regard to operating functionality. Nonetheless, it supports the execution of interactive NSW tools with completely distributed components.

The major portion of the work to produce a TENEX Foreman for a multi-host NSW environment was to adapt to and utilize ARPANET MSG. MSG supports the interprocess communication that couples the Foreman component with the NSW Front End and the NSW Works Manager. This involved providing a Foreman interface for sending and receiving MSG messages and alarms, detailing the nature of the resource allocation strategy to be used by MSG for handling the TENEX Foreman, and beginning to adopt the conventions and protocols for running NSW tools, as outlined in the Foreman specification document (BBN Report #3266).

In the current TENEX Foreman implementation, many tool instances, each with its own Foreman, share a single TENEX job and a single MSG supervisor process. The MSG facility creates (and ultimately deallocates) Foreman processes within this job as needed. Each Foreman, in turn, creates and monitors a process for executing the tool in question. Because of the flexible nature of the TENEX process structure, and as a result of recent TENEX system changes to help support such an organization, it is

possible to run multiple tools in a single TENEX job. We plan to compare this approach experimentally with the approach of creating a new TENEX job for each tool instance. We believe that the multiple-tools-per-job approach offers significant performance advantages. MSG allocates a Foreman process to each message generically addressed to a Foreman (on that host). The generically addressed message for the Foreman, which contains the name of the tool to be run, starts the chain of events and messages that places the user in direct communication with the tool.

The first steps taken to make the TENEX Foreman operational for a multi-host NSW were the implementation of generalized routines to use the MSG communication capabilities and to adhere to and parse the protocol agreed upon for the inter-component messages. Next, implementation of the BEGINTOOL function in the Foreman made it possible to run a TENEX NSW tool started by a remote process. By converting the NSW file retrieval and delivery functions to use a non-local Works Manager, we achieved an initial working TENEX Foreman. This Foreman, however, provided the bare minimum for functionally interacting with the other NSW components. The remainder of the quarter was spent working on a more complete implementation of the Foreman as specified in the Foreman design document.

Among the Foreman capabilities that became operational this quarter are the gathering and reporting of tool resource use

charges, and a mechanism for advising the Foreman that a local host system restart has occurred. Whenever a TENEX NSW tool terminates, its Foreman now totals the computer system resources used during the tool session and computes the cost of running that tool instance. These resource utilization figures and charges are reported to the Works Manager, which updates the user's accounting record accordingly and informs the user of the charges just incurred. On TENEX, we account and charge an NSW user for the time connected to the tool and for the CPU time used to run the tool. File storage used during a tool session is temporary and thought to be negligible. Hence, there is currently no charge or accounting for this resource. Long-term file storage (in the NSW file system) will, however, be directly charged on a usage basis.

Ordinarily, a Foreman process is invoked to provide tool execution in response to user commands. When a Tool Bearing Host is restarted after a system crash, the Foreman must perform certain crash recovery procedures. In order to avoid the loss of more files resulting from a TBH system crash, the Foreman component has the responsibility of maintaining the state of its tool's workspace in a "crash-proof" manner and, if the TBH crashes during a tool session, the responsibility of recovering as many of the user files as possible from the workspace after the TBH is restarted. The first step in supporting this file recovery procedure is to notify a Foreman that a TBH crash has occurred or, equivalently, that the NSW software on the host has

been restarted. We have provided such a system startup notification mechanism through TENEX MSG, and are currently using it to handle the Foreman job initialization. This initialization will be modified in the future to include file recovery operations for any tool workspace that was active when NSW operation on the host ceased.

Also during the past reporting period, the TENEX NSW tool encapsulation part of the Foreman was enhanced substantially to allow more TENEX tools to be used through NSW. As a result of these enhancements, the computer system emulation and debugging facility programs (PRIM, UYK20, U1050) developed at the Information Sciences Institute (ISI) of the University of Southern California are now operable as NSW tools.

Throughout the reporting period, we spent a substantial amount of time working with the other NSW contractors in system integration and checkout. We have found the integration of functional components for the NSW to be a much more formidable task than we had expected. We believe this is due, in part, to the complexity of the system and, in part, due to the lack of effective debugging tools and technique for distributed, multi-computer systems. Despite these difficulties, these system integration and component efforts resulted in the successful July demonstration of multi-computer NSW operation.

4. Other Activities

As Part of our implementation work, we upgraded the TENEX Front End Dispatcher to function along with TENEX MSG. In a multi-host NSW, the TENEX Front End must have an MSG module that oversees its operation and provides the MSG utilities. To achieve this, we have specified and implemented a Dispatcher-MSG protocol for dynamically establishing Front End processes in response to remote requests users. Part of this protocol includes a convention by which the Dispatcher can signal MSG when error conditions are detected, such as, broken network connections, so that the Front End process can be terminated in an orderly manner. Another aspect of this protocol deals with the deallocation of job resources used to run the Front End after the user has completed his NSW session.

During this quarter, we collaborated with Massachusetts Computer Associates to produce the preliminary version of the NSW Tool Builder's Guide (BBN Report No. 3308, COMPASS Document No. CADD-7605-2111). This is an important document for tool purveyors and builders who must interface their tools to the NSW. The preliminary Guide contains the overview and general NSW aspects of NSW tool building. Still to come are the portions of the Guide, which are specific to the various NSW Tool Bearing Hosts. This will include a section on building TENEX NSW tools.