DATE: January 14, 1982

TO: RD&E Personnel

FROM: The NMS Team

SUBJECT: The NMS Definition Document

REFERENCE: None

KEYWORDS: None

ABSTRACT

This is the first document in a series of Network Managment Services (NMS) literature. It supersedes all NMS documentation issued prior to this document's release date. In particular PE-TI-953 should no longer be considered definitive - it's replacement will be issued soon.

This PET outlines areas that it feels Prime will need to address within the next few years if it is to remain competitive in the Networking arena. These areas can all be considered under the one heading, "Network Management Services".

This PET also describes the way in which the project will be tackled and offers a glossary of terms that will be used throughout the projects documentation.

Please address any comments, criticisms and suggestions (for items to be included in the glossary) to Gary Freeman - X.Mail GARY on BED1 or 'phone the Bedford office, x244.

Table of Contents

1 INTRODUCTION....................3
1.1 Layout Of This Document....................3
1.2 Need For This Document....................3
1.3 Purpose Of This Document....................4

2 THE NMS ENVIRONMENT....................5
2.1 Motivation for Change....................5
2.1.1 New Processors....................5
2.1.2 New Architecture....................6
2.2 The Effects of Change....................7
2.3 Summary - What Is NMS?....................8

3 SCOPE OF THE NMS PROJECT....................9
3.1 Scope Of the Monitoring Phase....................9
3.1.1 Generalised Logging Subsystem....................9
3.1.2 Configurations Monitor....................10
3.1.3 Event Handling Subsystem....................10
3.1.4 Dynamic Monitoring....................11
3.2 Scope Of The Control Phase....................13
3.3 Scope Of The Diagnostic Phase....................15

4 PROJECT GOALS....................17

5 PROJECT STRATEGY....................18

6 INITIAL PLAN OF CAMPAIGN, RESOURCES AND TIMETABLE....................19
6.1 The Requirements Document....................19
6.2 The Functional Specification....................19
6.3 Investigation Of The Problems....................19
6.4 Full Implementation: Design And Building....................19
6.5 Release Documentation....................20
6.6 Resources And Timetable....................20

7 SUMMARY....................21

8 GLOSSARY....................22

1 INTRODUCTION

1.1 Layout Of This Document

This document is in 8 sections.

This section explains the need for this document and its purpose. Subsequent sections are as follows:

Section 2: Outlines the trend, within the industry, towards networking and indicates how Prime will be producing new hardware and software in order to stay with that trend.

Section 3: Outline of the three areas of NMS and their respective scopes.

Section 4: Overall project goals.

Section 5: The Strategy that will be adopted to implement those goals.

Section 6: The "Plan of Attack" for implementing each of the phases.

Section 7: Summary.

Section 8: Glossary.

1.2 Need For This Document

Documentation for this project has, so far, followed the traditional path of Prime project documentation: a discussion document, a project plan, a functional specification and so forth. This approach is limited in that it encourages authors to group together information that should really be held in discrete documents. For example, functional specifications often contain requirements and philosophies in addition to "functional specifications". This can lead to some information being implied or misunderstood and, ultimately, an unwanted subsystem being developed.

A new approach is now being adopted by several projects (e.g. "Forms Mode Terminal Services Project") whereby the documentation is split up into a more logical pattern. Each paper contains only one type of information and consequently is easier for authors to write and reviewers to criticise!

This approach to documentation has now been adopted by the NMS project and this is the first such document.

1.3 Purpose Of This Document

This document supersedes all previous NMS documentation. It's purpose is to broadcast to the Engineering community the existence and activities and philosophies of the NMS project. It is a "Big Picture" of the overall project without getting down into the details.

This document aims to make it quite clear:

- what we are doing;
- what we are going to do;
- why we are going to do it at all.

It is intended to stimulate thought and discussion. Please feel free to discuss this paper with others and, most importantly, please let us know of your conclusions.

In particular:

What appears to be omitted from this project that shouldn't be?

Where does this project duplicate other project's efforts?

What is wrong with this project?

We guarantee a reply to any and all input that you may have.

Throughout this document examples are freely cited as to what NMS might contain. Such examples do not indicate commitment for their inclusion to the NMS project. They will only be included if both marketing and engineering acknowledge that there is a need for them. Section 6 describes how this process will take place.

2 THE NMS ENVIRONMENT

Significant changes in the type of system that PRIME markets are imminent. The changes will bring about additional requirements of Prime software from users. This section outlines the motives, and the likely direction, of such changes.

2.1 Motivation for Change

The general motivation for change is the increasing number of Prime machines that are being sold as part of a network. This is being brought about by the comprehensive connection capabilities which allow Prime machines to form part of both Wide Area Networks (WANs) and Local Area Networks (LANs).

More specifically, there is a two fold stimulus that will increase the rate of change towards networking: the introduction of CPUs specifically designed to be used as part of a network and a new systems architecture designed once again, with distributed systems in mind.

2.1.1 New Processors

Prime CPUs were originally sold individually; to customers requiring support for a large number of interactive users. The need for several Primes to be connected as a LAN came primarily from customers whose requirements exceeded a single machine's capabilities. The need for WANs came from larger customers requiring (many) geographically dispersed nodes to be connected together.

Although these two types of customers still exist, subtle changes have occured in both LANs and WANs. A node in a LAN is now expected to provide a small user community (perhaps a single office or a typing pool) with some dedicated processing power which will remain unaffected by the failure or temporary shutdown of other nodes. At the same time each node is expected to be able to provide access to non-local databases. These LANs may exist autonomosly for most of the time (e.g. at a branch office), however, periodically, they may become part of a WAN and be connected to a central site (perhaps to update a central database) via a "gateway".

It is unlikely that staff attending one of these small machines will have any operations knowledge. They will simply turn the machine on and use it.

Prime's first step towards environments of this type lies with RABBIT (a P250 sized machine housed in a cabinet suitable for an office environment). This machine incoporates a new, intelligent communications controller which supports 8 asynchronous lines (typically for terminal support) and one synchronous line for connecting to a network. This machine will be marketed not as a node in isolation but as a node for use within a network (either LAN or WAN).

2.1.2 New Architecture

Prime is not acting alone when it advocate the increasing use of networking. Indeed, the International Standards Organisation (ISO) believe that networks should not be restricted to containing nodes of a single manufacturer. To this end they have developed an architectural reference model for Open Systems Interconnections (OSI). This model defines a framework for transferring data between nodes dissimilar both in terms of model and manufacturer.

Prime's Distributed Systems Architecture Group (DSAG) have studied this standard and are using it as input to their design of a new architecture which will help take Prime into the 1990s. Within this architecture users request "services" (e.g. a compilation, a file or a peripheral). It is considered irrelevent where on a network these services reside.

For example, consider a LAN of several nodes. When a user requests a compilation of a Fortran program, several services will accessed: the Fortran compiler itself, the allocator of file system space and the files themselves. The system will choose which copy of the Fortran Compiler to invoke according to several parameters (e.g. whether or not the compiler resides on the same node as any of the files and hence be able to reduce tne number of remote accesses to remote files; which machine is lightly loaded at present; which machine has "hardware assists" to help speed up compilations and so forth).

This contrasts greatly with todays environment where each node is a "Jack of All Trades" and users are restricted to selecting the service residing on the node to which they are logged in.

2.2 The Effects of Change

Even with the advent of increased "low level" networking capabilities, customers are still restricted to using "high level" tools made available for the single node environment (e.g. ELIGTS, USAGE and PBHIST). These are inadequate for the administration of networks as they provide only limited information/ control over a single node and cannot aid in the monitoring or controlling of data flow between nodes. Additional tools must be provided to enable customers to monitor and control their networks with confidence.

"Monitoring" and "Control" tools are not required simply to handle problematic networks. They are fundamental in ensuring the smooth and efficient day to day running of a network. USAGE can only identify which users are consuming large amounts of a given resource, it can not show what the user was doing with that resource. Such information is invaluable as an aid to scheduling resources. The need for such information is compounded in networks where several nodes are vying for each others resources.

Further, when things do go wrong we must adopt a better approach to informing users. The "error message to the Systems Console" concept is no longer appropriate.

In LANs of say ten machines there are ten consoles. Operations staff can not be expected to periodically scan these for error messages. The problem is highlighted if the "system in error" resides on a node unattended by computer staff (as in the branch office example described earlier) - the message may never be seen!

The two areas, "monitor" and "control", exist in both the single and multiple node environments alike. It is only the degree of sophistication required in such tools that differs between these environments. Once the monitoring tools have improved and the customer is more aware of a network's workings he will rightly require equally sophisticated tools with which to control that network. Tools to manipulate traffic flow between nodes; to schedule and re-schedule resources between nodes and so on and so forth. Ultimately the network itself should be capable of such manipulation but until that time comes the customer will need tools which enables him to achieve this end.

An area not yet considered (and specific to networking) is that of communications diagnostics. The need for diagnostic tools exists even when the connected nodes are of the same manufacture because quite often the interconnection between them has elements provided by a third party (e.g. modems or a packet switching service). The need is emphasised when the connected nodes are of different manufacture, for should a problem occur the representatives of both manufacturers may point the finger at each other! It will become essential for our support analysts to be able to prove the extent of their responsibility and this will only be possible if they have the correct equipment to do so.

2.3 Summary - What Is NMS?

Despite PRIME's (and PRIMOS') expanding capabilities, administrators have very few tools to aid them with their control of the network. Not just the "controlling" tools themselves but also the monitoring tools which would enable administrators to see the effects of their actions. These tools would be used in normal day to day running of a network and not just when things go wrong.

When things do go wrong most subsystems cope quite well and either report things to the systems console or to a log file. One facility provided by NMS will make it possible to report errors to specific terminals (and hence provide the administrator and not the systems console with the information). More details of the tools will be provided and can be found in section 3.

Although many subsystems can identify problems at a "high" level, it is sometimes difficult to identify the low level faults. This is particularly true in the communications environment where not only the faults but also the offending node (or component) is difficult to isolate. NMS aims to provide such diagnostic tools.

NMS, then, is a selection of Network Administrative tools. NMS considers a CPU to be a degenerate network; and with it all networks can be controlled in a simple and compatible manner.

The three areas of NMS: Monitor, Control and Diagnosis will be described in more detail in the next section.

3 SCOPE OF THE NMS PROJECT

This section describes the three phases of the project and highlights the areas that NMS will be considering.

3.1 Scope Of the Monitoring Phase

The Monitoring phase of NMS will provide users with a variety of tools with which to monitor a network. Tools to indicate problems, to show loading, configurations and bottlenecks. Not only interactively but also offline, generating reports which can help administrators identify their future needs and current problems.

Four main categories are envisaged:

- o Generalised logging of system and subsystem activities.
- o A subsystem to display current system and subsystem configurations.
- o A subsystem to track and report unexpected events.
- o A monitor subsystem to interrogate the machine for statistics (e.g. "usage" type information)

Consider each of these four areas in more depth.

3.1.1 Generalised Logging Subsystem

Logging is an important part of any subsystem. It enables both diagnostic information and statistics to be easily recorded. Usually the data is written to a file and interpreted later either by the PRIMOS Editor or by a utility tailored to the specific file format. Many projects have defined a logging mechanism and this has resulted in "the wheel being invented" time and time again.

This part of the project will involve indentifying the need for a general purpose logging mechansim. If such a mechanism proves necessary the NMS team will identify its requirements both in terms of recording the data and interpreting it at a later date; and then implement a facility to meet these needs.

3.1.2 Configurations Monitor

Some aspects of PRIMOS can be configured dynamically. For example the line characteristics (using the AMLC command), the maximum number of users (using MAXUSR) and so forth. Unfortunately, it is not possible to obtain the current values that have been set by such commands. Once again, an investigation into the need for such tools will take place and, where necessary, they will be developed.

Also, in this part of the project, an invesitgation will take place to establish the need and practicality of "standardising" configurations display formats for various PRIMOS subsystems.

3.1.3 Event Handling Subsystem

Subsystems, when detecting problems that they cannot overcome, usually adopt one of three approaches:

- Report the problems to the systems console
- Report the problem to a log file
- Don't report the problem!

These approaches can sometimes be defended, however, they do have the following drawbacks:

- The systems console is used by customers less frequently as a central site for collecting information. This is especially true in local networks where there are as many consoles as there are nodes on the network. In such environments events reported to the systems console are not noticed as swiftly as they should be.
- Similarly, a log file is not studied frequently enough for some problems to be dealt with in time.
- (The reader is invited to write polite comments here!).

Marketing has already identified a requirement for a facility that will enable subsystems, that have detected otherwise unmanageable events, to notify the appropriate administrator of the event. The manner in which this this is achieved will be installation dependant.

Options currently under consideration are:

1. Provide dedicated VDUs whose sole purpose is to display the occurence of a particular type of event. It will be possible to configure the type and origin of events that are displayed at each of these VDUs.

2. Upon detecting an event the subsystem will send a message to the terminal of a specified user (e.g. network manager or operator) or systems console. Note: this approach does not remove the need for a systems console but does aid in an overall plan to make it optional).

3. Mailing a specified user.

Note: In addition to points 1-3 it will be possible to keep a log file of all events raised (using the generalised logging subsystem mentioned in 3.1.1) for interpretation and analysis at a later date.

3.1.4 Dynamic Monitoring.

The extent to which PRIMOS resources fluctuate (e.g. CPU loading and packet flows throughout a network) determines the way in which the network appears to individual users. For example: if two geographically remote ring networks are to be connected together a method frequently used to achieve this is one whereby a node on each ring is designated a "gateway". These gateway nodes are connected together to form a WAN. All traffic travelling from one ring to the other must pass through these gateways. Even though the requests, from an individual node, for data transfer may be quite light, the cumulative requests from many nodes may force a gateway into saturation. I.e. it becomes a bottleneck. But how does a user, whose request for a data transfer is taking an exceptionally long time, know where the problem lies? The delay might easily be the result of:

any one of the CPUs through which data passes being CPU/IO bound

one or both of the rings themselves being heavily loaded

the gateway nodes being overworked.

This list is not exhaustive, but it does show that network monitoring requires more thought than is needed to monitor a single node. Also, it is apparent that a variety of monitoring aids are required to monitor different aspects of a network. Which aspects need to be monitored will be decided upon after investigation. However, the following will be used as a starting point for our investigation:

o CPU monitors (similar to USAGE) but with more flexible display formats and with output options suitable to terminals such as the PT45 and WREN.

- Network Traffic Monitors - indicating packet transfer on a per node, per user or per port (subsystem) basis.

- Network queue monitors at both the high level (eg BATCH, RJE or SPOOL queue) and lower level (RJE internal queues, PRIMOS queues, Semaphores, IPQNM queus and free pool queues).

If you know of any additional system aspects that you feel need monitoring then please let us know.

3.2 Scope Of The Control Phase

The Monitoring tools provided by NMS may indicate the need to alter one or more of a network's aspects. This may involve redefining either a subsystem or the Operating System's configuration itself.

Consider first the subsytems: currently there is no standard to define the way in which a subsystem differentiates between a user with administrative rights and one without them. The most widely adopted approach is to recognise the "SYSTEM" user as the only one with administrative powers. A more flexible approach is anticipated with the advent of ACLs: e.g. each subsystem could allow administration rights to specific users or user groups (indeed this approach would also allow subsystems to validate requests from users on remote nodes when used with NPX). However, as yet, there exists no command line convention (other than full treename specification in the RESUME command) to indicate that the requested subsystem resides on a node other than the local one.

Such a feature could be useful in a variety of applications. For example on rings where several printers are distributed throughout the network (each serving one form type in several machine's spool queues) it would be useful if a user with several files spooled on his local machine could check to see if the remote despooler was running. I.e. a "PROP -STATUS" for a specified node. Similarly there is also a need for such a facility on FTR, BATCH and numerous other subsystems.

At the command line this functionality could be achieved with a "-ON" option e.g.:-

PROP -STATUS -ON ENB

Although this is straightforward at the command level, it is not quite as easy to implement especially as the security aspects in such an environment are the most important features of the system.

Whereas it would be advantageous to have the "-ON" option for user commands, it is essential that such a mechanism is made available for "User 1" commands. Not only for configurations that have a central site from which geographically remote nodes (unattended by computer staff) may be controlled. But also in large ring networks where we no longer wish to see an 'N' node system requiring 'N' System Consoles. In this latter environment the future lies in being able to take control of entire networks from any terminal on that network.

The concepts discussed so far describe networks that have no node failures. When a node does go down it should be possible to gain VCP control of that node, again, from any terminal on the network.

To summarise:

During the Control Phase of the NMS project we will investigate the requirements of, and implement, the necessary tools which will enable an administrator to:

1. Assume 'User 1' control of those nodes on a network for which he has appropriate rights, from any terminal on the network.

2. Take "VCP" control of those nodes on a network for which he has appropriate rights, from any terminal on the network.

3. Take administrative control over any subsystem on a network for which he has appropriate rights, from any terminal on the network.

Note: A side effect of this Phase will be the determination of the desirability and feasibility of a generalised form of '3', above. I.e. To have decided how one might implement a secure method of allowing inter node invocation of any PRIME software. Example of such usage might be:

SPOOL FRED -ON BED6

By adopting this approach, the copying of the file from the local to the remote machine will take place at 'spool time' (i.e. from the local file to the remote spool queue) and not at 'despool time' (i.e. from the local spool queue to the remote printer). Consequently, the printer will not find itself waiting for traffic coming across the network and hence may be driven more efficiently.

Another example might be:

FTN <NMSDSK>PROGS>TEST.FTN -ON BED6

In this example the user would be able to take advantage of his knowledge that <NMSDSK> is local to BED6 and (by specifying the node) reduce the traffic across the network. Admittedly, environments such as those being developed by the DSAG team should be able to make such decisions without user aid but until then this type of option should be useful.

3.3 Scope Of The Diagnostic Phase

Although individual nodes and their subsystems are subject to failure, it is assumed that they are provided with their own debugging aids which can identify problems when the occur. The NMS Diagnostic Phase will NOT assist further in these areas. Its purpose is to identify the faults (and where they lie) between two or more nodes on a network. Problems of this nature are difficult to isolate. The 'Symptom' of a fault can often be caused by a variety of errors (in either hardware or software) and so the symptom itself does not automatically identify the problem. Tools are required which will systematicaly reduce (usually by a process of elimination) the time to isolate the REAL problem.

The types of faults that the tools of this phase should be able to detect are varied. To simplify matters, consider first a simple network of two Prime machines connected by a synchronous link:-

The machines in this example are connected to each other via a 'phone line. Each node(N) is connected to the line with a Modem(M) and, diagramatically, can be thought of as follows:-

```
 -----                                                      -----
|  N  |----M-------- - - - - - - - - - --------M---|  N  |
 -----                                                      -----
```

In this type of an environment, faults may occur in a number of places:-

In the networking software on either machine.

In the Inter Process Communications (IPC) mechanism between the networking software and that on the comms. controller.

In the comms. controller itself (hardware or software).

On the wires connecting the controller to the modem.

On either modem (Hardware fault, incorrectly configured)

On the wires between the modems

A fault in one of the above areas will often result in a total lack of communication. These problems are usually easiest to resolve by a process of elimination (eg. testing parts of the network in isolation).

With this in mind, tools which might be useful include a modem 'controller', whereby an administrator may switch either the local or remote modems into loopback (from his terminal) and then send CCITT test patterns down the line and compare them with the incoming patterns that have been looped at one of the components on the line. By altering where the loopback takes place (eg at the local controller, the local side of the local modem, the remote side of local modem, the local side of the remote modem, etc etc ....), the administrator would be able to identify where the fault lay. The ability to switch modems into loopback, by sending them signals on their secondary channel, has only recently been made available and a lot of research will have to be made in this area.

When the communication's problem is intermittent, the above approach will not always yeild results. Admittedly the problem could be of the 'dry joint' type, in which case it might be identified by a 'process of elimination'. However many intermittent faults are caused by the content of the data packet/block (not the media through which it is travelling) and hence the previous approach would be inappropriate. A tool which would help here would be one that stores the dialogue between two nodes in a cyclic buffer and when a problem does occur the stored data would no longer be overwritten and it would be made available to the administrator so that he could identify the offending data. With a tool of this nature the problematic link could be continually monitored for the intermittent fault without an analyst sitting at a datascope for hours on end waiting for it to happen.

Whether or not other Datascope like features would also be adopted will depend upon the Marketing requirements and the ease with which they could be implemented.

4 PROJECT GOALS

This project aims to provide (for all users of a network) tools that are:

- inkeeping with the requirements of the user (as defined in the requirements document - see section 6).
- robust;
- user friendly;
- well documented;
- easily maintained and enhanced;
- compatible with existing PRIME conventions;
- designed to be easily upgradeable to the DSAG environment;

5 PROJECT STRATEGY

To achieve the project goals, listed above, we have adopted the following initial strategy:

- To identify all aspects of the requirements and issues of the service which we aim to provide.

  These will be presented in the project requirements specification;

- To generate requirements for other projects

  In all three phases (Monitoring, Control and Diagnostics) there will be areas of activity that are beyond the scope of the NMS team (eg enhancements to the Comms. controllers). The NMS team will identify these areas as swiftly as possible and supply detailed requirements to the appropriate project teams as necessary.

- To announce and advertise the functional aspects of the proposed system.

  We will require feedback on our proposed implementation approach and will also need to be clear in our own minds that we have chosen a viable strategy: - a sort of "walking before we run" approach. To this end, we will publish a functional specification.

6 INITIAL PLAN OF CAMPAIGN, RESOURCES AND TIMETABLE

6.1 The Requirements Document

Following this paper, work will proceed on a Requirements Document for each section of the NMS project. The first one will be for the Event Handler section of the Monitoring Phase.

Each document will state specifically the joint requirements of both Marketing and Engineering. If a feature is not mentioned in this document then it may be assumed that it will not form part of this project. Representatives of both Engineering and Marketing will put their names to this document to confirm that it accurately defines PRIME'S requirements of the project.

It will not be concerned with the functionality of the tools: i.e. what the tools will provide in terms of interfaces or functions. Neither will it be concerned with how the tools will provide the required functions. These matters will be treated separately by the functional specification and by implementation design documents.

6.2 The Functional Specification

The functional specification will contain proposals for solutions and approaches to the problems identified in the requirements paper. This document will describe the tool's external appearance and explain the rationale for the approach adopted. This document will not be concerned with how they work.

6.3 Investigation Of The Problems

Once the functional specification has been reviewed and agreed, we should be in a position to try out various ideas to test our approach and look for conflicting side-issues and dependencies. We will experiment with non-marketable prototypes where necessary.

6.4 Full Implementation: Design And Building

This should follow on naturally from our previous work and should result in a phased release of tools which provide the required functionality and satisfy the project goals.

6.5 Release Documentation

When a tool is released it will contain documentation within the source files themselves. Where the volume of information would represent a burdon on the file system, PE-TIs will be supplied describing the internals for specific tools and the rationale behind the approach chosen.

6.6 Resources And Timetable

With any project, resources and timetable are subject to frequent adjustment. Therefore, there is little advantage in presenting this type of information in a PE-TI. However, such information is maintained on line and can be accessed via reference to the author.

7 SUMMARY

This document has described an area of computing which has not yet been tackled by Prime: that of being able to consider multi-node networks with the same ease with which single machines can be considered.

It has shown that the concept is one that Prime does need to address before it expands into Markets which require a more sophisticated approach to networking. Further, these enhancements will improve the system as seen by administrators of single nodes.

There are 3 main areas for improvement: Monitoring, Control and Diagnosis. This document then suggests (without commiting the project to implementing) some features that may be included within each phase. Finally, it concludes with a description of how the exact details of the project will be defined.

8 GLOSSARY

/******* Please suggest which terms you *******/
/******* would like in this section *******/

SERVICE
SERVICES
MONITORING
CONTROL
DIAGNOSIS
NETWORK MANAGEMENT
LANS
WANS
DISTRIBUTED
DE-CENTRALISED
NODE
GATEWAY
ASYNCHRONOUS
SYNCHRONOUS
COMMUNICATIONS CONTROLLER
MODEM
NETWORK/SYSTEMS ADMINISTRATOR
PACKET
REMOTE NODE
LOCAL NODE
VCP
CCITT
DATASCOPE