NetView/PC

by M. Ahmadi
J. H. Chou
G. Gafka

NetView/PC™ is an IBM program offering that provides the first implementation of a Systems Network Architecture (SNA) service point on an IBM Personal Computer. It allows integration of non-SNA devices such as computerized branch exchanges or Token Ring local-area networks into a central SNA network management facility by forwarding Alert information from these devices to a host-based network management product such as NetView™. It also provides some focal point services, including logging and display of Alerts and problem creation and tracking. Non-IBM products may use NetView/PC services via a vendor application programming interface (API).

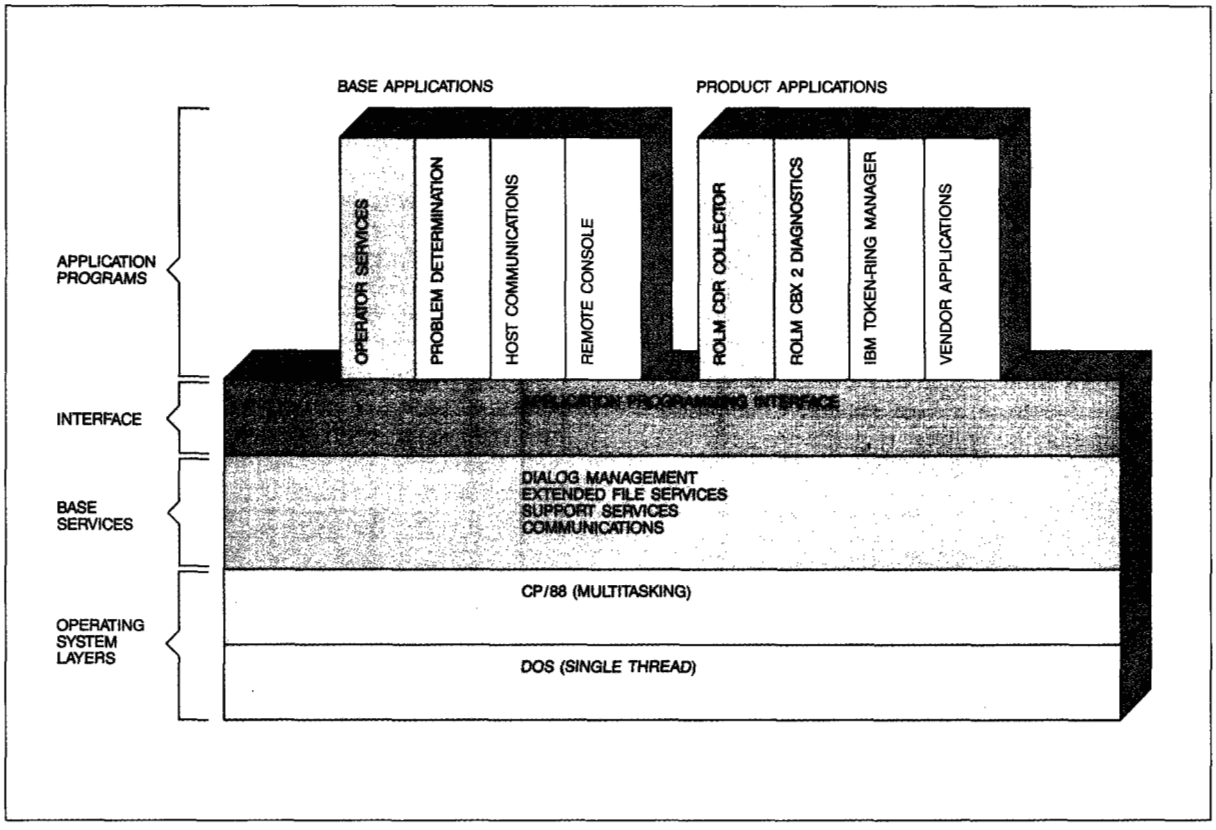
IBM recognizes the difficulties encountered by customers who need to manage large networks, especially networks composed of products from different vendors. In order to allow customers to achieve "end-to-end" network management, IBM continues to enhance Systems Network Architecture (SNA) and continues to enhance it in order to keep pace with increasing customer network requirements.

SNA. SNA provides a framework for network management. The network management architecture of SNA defines three network management product applications: focal points, entry points, and service points. A focal point is a product that provides centralized network management support. This support includes the storage, retrieval, analysis, routing, and display of diagnostic information. Entry points are SNA-addressable products that provide connectivity for downstream SNA-addressable network components and that forward formatted network management information to a focal point for processing. Service points are products that provide a connection through which network management information from non-SNA devices may be sent to a focal point. The service point itself is SNA-addressable and can transform SNA-formatted network management requests from a focal point, translate them, and forward them to the appropriate non-SNA network. The same services can also be performed in the reverse direction.

NetView/PC. NetView/PC™ is a program product which runs on an IBM Personal Computer XT or IBM Personal Computer AT® (IBM PC XT or IBM PC AT) with a fixed disk and 640K of memory; a Real-Time Interface Coprocessor (RTIC) adapter card is also required. The operating system required for the NetView/PC 1.0 program is the IBM Personal Computer Disk Operating System Version 3.2 (DOS 3.2) or greater.

The NetView/PC program provides both service point and focal point functions as defined by SNA. As a service point, it provides a method for non-SNA devices to communicate with an SNA host. As a focal point, it provides network management functions, including error notification, problem tracking and management, and work list tracking. In addition to these functions, NetView/PC provides a computing platform with a number of services for IBM and vendor application programs.

Figure 1 Components and structure of NetView/PC



Service point functions. The NetView/PC program may act as a service point in a large SNA network. Information from devices such as computerized branch exchanges (CBXs™), token rings, T1 multiplexers, and so on may be gathered by an application program, translated from the format of the IBM PC to that of the System/370, then forwarded to a host-based network management program such as NetView™. This procedure allows all vendors to participate in centralized network management and enables customers to manage both SNA and non-SNA devices from a single facility.

Focal point functions. In cases where a host is not available or not desired, NetView/PC can also function as an SNA focal point. NetView/PC provides a number of focal point functions, including error notification, logging and display of Alerts, and a problem management facility which enables the creation and tracking of problem reports.

Structure

The overall structure of NetView/PC is shown in Figure 1.

The major components of the NetView/PC program are the base system, communications support, NetView/PC applications, and product applications. IBM applications, including both the NetView/PC base applications and IBM or ROLM® product applications, access base system services via an IBM internal application programming interface (API). A vendor API is also provided to enable non-IBM applications to use these services.

Base services. The NetView/PC program runs under CP/88, an IBM-developed operating system for the IBM PC. This operating system provides multitasking, allowing many applications to be run concurrently. Most DOS and BIOS (Base Input/Output System) calls

are trapped and either serialized or emulated. CP/88 also provides support for multiple sessions, each session consisting of a logical display and keyboard which may be used by the application running in that session.

CP/88 was used on top of DOS because multitasking was a requirement for NetView/PC. A number of different processes must run concurrently in order to provide all NetView/PC services. If an Alert (a message generated by a hardware or software product when an error is detected) is received, for example, it must be checked for errors, sent to a host, and placed in a database. While one Alert is being processed, another may be in the process of being sent to a host. At the same time, NetView/PC may also be receiving a file from a host system, or a product application may be polling its hardware devices. In addition to all of this, a user may be using the focal point services to check some information.

In addition to multitasking, the base system also provides dialog management, structured file services, and a number of other support services.

Dialog management. All NetView/PC base applications that require a user interface use a re-entrant version of a dialog management product named EZ-VU. This version is not available to customers. EZ-VU provides many of the functions found in the Interactive System Productivity Facility (ISPF), a host dialog management product, including validation of entry fields, panel display, checking of function keys, and so on. The use of EZ-VU increases the consistency of the user interface, thus increasing the usability of NetView/PC.

File services. In addition to the standard DOS file I/O functions, NetView/PC provides extended file services to applications. These services include support for linked-list files, sequenced files, and log files. The file types basically refer to the methods by which records are stored and accessed. Different NetView/PC base applications require different database types, but since there was some overlap, database support was provided as part of base services rather than placing the burden on the applications. Basic database functions are provided, including file creation, deletion, and archiving, and record insertion, retrieval, modification, and deletion. Some search functions are also provided.

Support services. In addition to dialog and file services, the base system also offers a number of support services to application programs. For instance, a name table is maintained which allows programs requiring interprocess communication to find the process ID (identifier) associated with another program.

Status line. Another important service which is offered is status line support. Line 25 in every session is reserved for status; whenever the status line is updated, it is refreshed in every session. The status line serves two major functions. One function is the display of exception condition messages that must be handled by the user; for instance, if a log file is overwritten, the operator is notified so that corrective action can be taken. The other major function is display of icons. Icons are basically two-character fields used to remind an operator of an event, or to present the status of an event. For instance, as shown later in Figure 4, the two icons on line 25 indicate that new alerts have been received (the AL icon) and that the link to the host is up (H↑).

Communications support. In order for NetView/PC to provide network management as a service point, it has to be able to communicate with SNA and non-SNA devices. For this reason NetView/PC provides the SNA Synchronous Data Link Control (SDLC) for host communication, along with asynchronous communication which is used for a remote console facility and for communicating with non-SNA and non-IBM devices and CBXs.

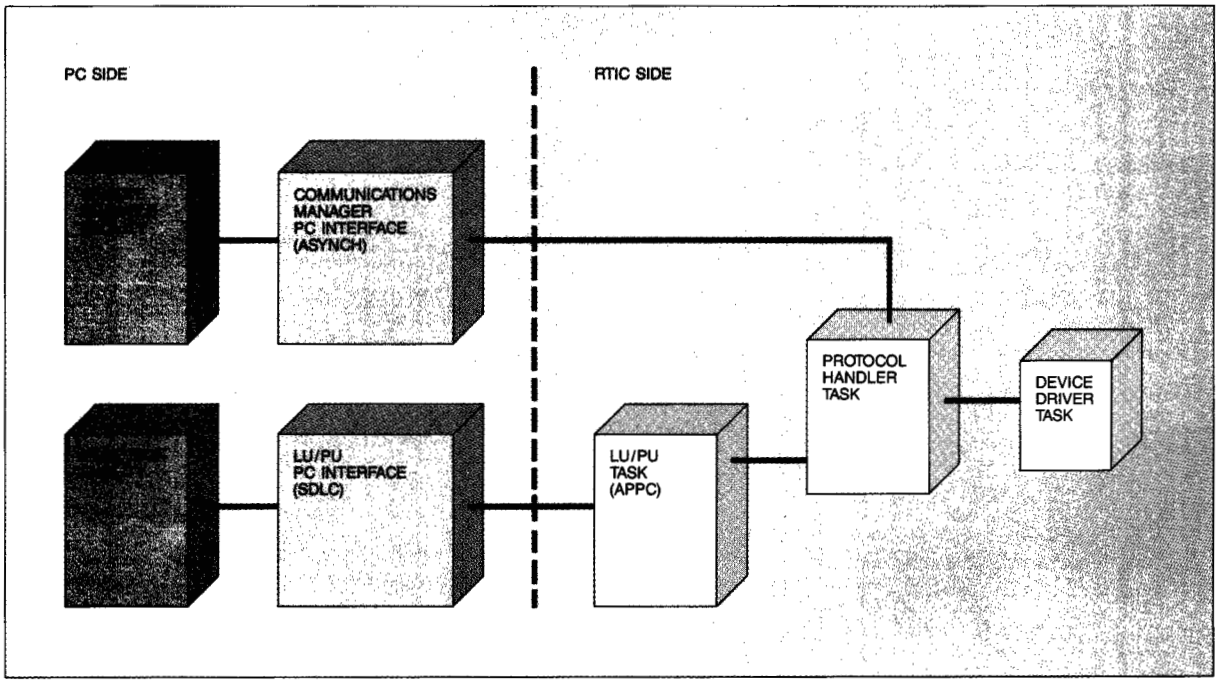
The SDLC communication operates in a half-duplex mode and can run at baud rates up to 9600 bits per second. Other options supported for SDLC are non-return-to-zero (inverted) recording (NRZI) and auto-dial for switched connection to the SNA network.

The asynchronous protocol supports odd-even parity, one or two stop bits, half or full duplex, and baud rates up to 9600 bits per second. Auto-dial and auto-answer are supported for asynchronous communication.

Figure 2 shows an overview of the communication components in NetView/PC. Three main components make up the communication support, and they reside and execute on the Real-Time Interface Coprocessor (RTIC). They are accessed via two separate interface tasks which execute within the IBM PC.

The LU/PU (logical unit/physical unit) task (also known as the Advanced Program-to-Program Communication, or APPC, task) provides the SNA PU 2.0

Figure 2 Overview of NetView/PC communications support



support for those NetView/PC functions that use the SSCP-PU session for host communication, such as the Host Alert Facility and Service Point Command Facility. Additionally, it provides the APPC on the LU 6.2 session. Currently, only the Host Data Facility of NetView/PC uses the LU 6.2 session for communication with the Customer Information Control System/Device Descriptor Module (CICS/DDM) on the host. The NetView/PC functions that use the APPC task access it through the LU/PU IBM PC interface.

The protocol handler and the device driver tasks are collectively known as the communication manager. The communication manager provides the link-level protocol support for asynchronous and SDLC communications.

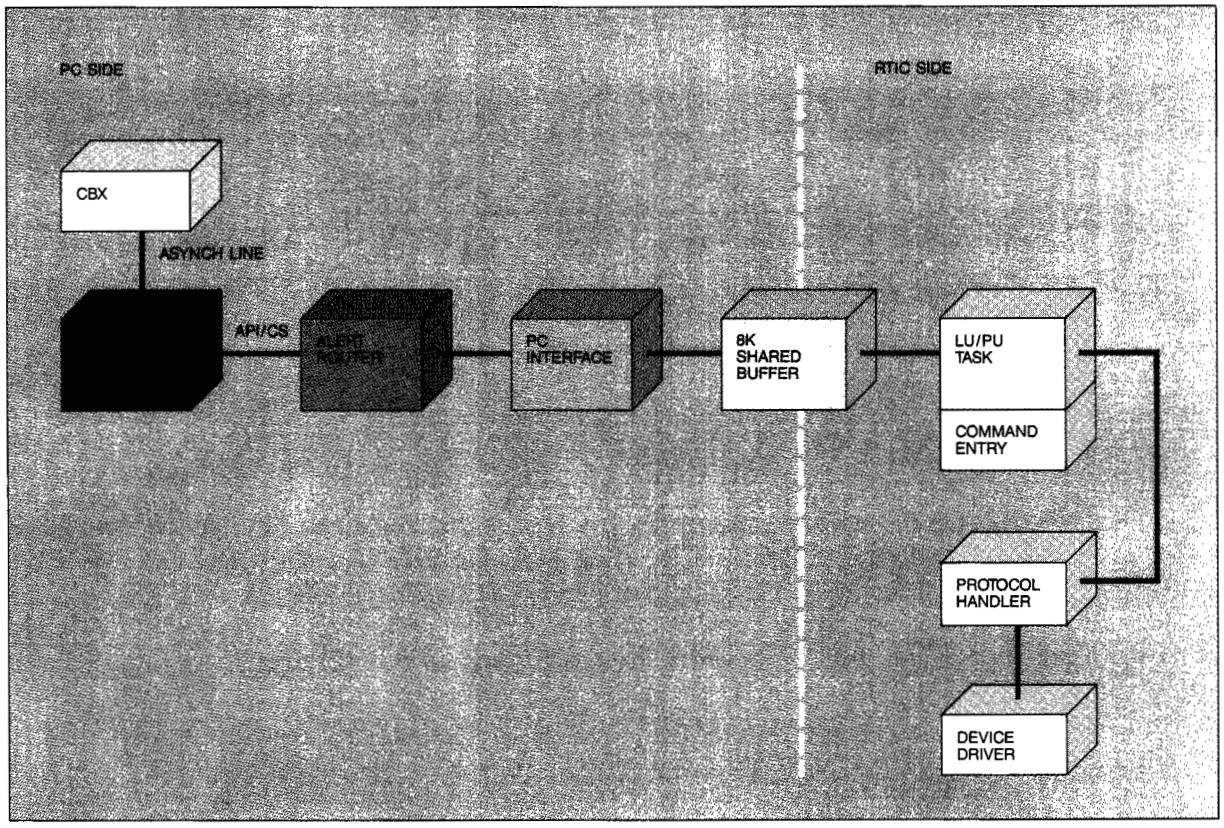
Any NetView/PC facility that needs asynchronous communication (i.e., remote console) accesses the communication manager directly through the communication manager IBM PC interface.

The Real-Time Interface Coprocessor. As mentioned before, the communication components of NetView/PC reside and execute on a coprocessor. There were two major reasons for putting the code on a coprocessor. The first reason is the IBM PC DOS storage limitation of 640K bytes, which prevents the communication code from being put on the IBM PC along with all the other NetView/PC functions. The second reason is performance, which has been improved by off-loading from the IBM PC processor all of the I/O activity that goes on in NetView/PC. The coprocessor also provides additional storage, which the LU/PU task uses to buffer the communication data. For example, in case of Alerts, the user application can send Alerts as fast as their application wishes. The Alert data are buffered in the coprocessor storage and sent asynchronously to NetView. In this manner, the user applications can continue to monitor their devices and do not have to set aside a large number of buffers for collecting their data.

The Real-Time Interface Coprocessor (RTIC), which was developed in IBM's Boca Raton, Florida, laboratory, provided NetView/PC with the desired storage and performance.

The RTIC is a single-slot full-length board which plugs into an IBM PC. The processor used is a 16-bit 80186 chip operating at 7.37 megaHertz. The RTIC uses

Figure 3 Structure of NetView/PC APPC support



120-nanosecond RAM (random access memory) chips and can support up to 512K bytes of storage. It also supports four types of electrical interfaces, of which NetView/PC uses the RS232-C interface.

The Real-time Control Program (RCP), also developed in Boca Raton, is the operating system for the RTIC. The RCP is a multitasking operating system that can control up to 253 concurrent tasks. Each task can execute on any of the 255 possible priorities.

The RCP also provides and manages such things as first-level interrupt handlers (both software and hardware), software timers, user queues, memory, communication ports, direct memory access (DMA) channels, etc. The data communication between the IBM PC and the RTIC is done through a shared-storage interface chip. This gate-array chip, which uses Complementary Metal Oxide Semiconductor (CMOS) technology, allows an IBM PC application to select any 8K-byte page of RAM on the RTIC, which the application can then write to or read from.

To further illustrate the communication between IBM PC and RTIC tasks, let us look at an example (see Figure 3). In this example the user application running on NetView/PC is monitoring a CBX. When the application detects a problem with the CBX, it will generate an Alert to be sent to NetView. In this case, the application, using the application programming interface/communications services (API/CS) routines, forwards the alert data to the ALERT ROUTER task. The data destined for APPC are given to the PC/Interface task by the router. The interface task will first reserve the 8K data window between the IBM PC and RTIC. This is done to prevent other tasks from writing to the RTIC storage at the same time. At this time the 8K-byte buffer is gated, and the Alert data are written to the RTIC storage. The interface task will then issue an interrupt for the LU/PU task (APPC) on the RTIC. The interrupt is handled by the RCP. The RCP gives control to a command entry point (second-level interrupt handler) within the APPC task. This interrupt handler puts a request on the queue of the APPC main task. At this point the second-level interrupt handler marks the window NOT_BUSY. At

the same time the PC/Interface task releases the window, so other tasks can use it. When the APPC main task is dispatched by the RCP, it removes the request from the queue and gives the data to APPC (in the form of an APPC verb). The APPC task then builds the Request/Response Unit (RU) and puts it on the queue of the protocol handler task. The protocol handler and the device driver tasks will then handle the actual sending of the data on the SDLC line.

Note that the user application is in a wait state only until the data are accepted by the APPC task. The PC/Interface allows the router and the user application to continue processing, once the APPC verb is accepted.

Communication enhancements. Some functions were added to the communication functions in NetView/PC to improve the usability and serviceability of the product. Following is a brief description of three of the main usability functions.

- **Host communication trace function**—The trace function allows the user to monitor all of the activity in the host communication function. There are two types of trace activity, PC (personal computer) trace and RTIC (coprocessor) trace. The PC trace, as the name implies, resides and executes in PC memory. It is mainly used by NetView/PC functions that provide host communication and is accessed by a macro interface. The type of information that is traced depends on the function. For example, the APPC interface manager monitors all of the verb traffic to and from APPC. If any of the verbs fail for any reason, the interface manager will trace the verb with its associated data. Other conditions that the interface manager will trace are activation/deactivation of the host session, activation/deactivation of an APPC task, and any communication error reported for the SDLC line (i.e., acknowledgment error, time-out). Other functions within NetView/PC trace different types of information or error conditions.

  All of the PC trace information is directed to and processed by a separate function called the trace manager. To activate the trace function, the user just loads the trace manager along with other NetView/PC tasks. When the trace macro is issued by a NetView/PC task, it checks to see if the trace manager is loaded. If it is, the trace data are forwarded to the trace manager. If it is not loaded, the trace call is treated like a null function.

  When the trace manager receives the trace data, it builds a header for the data. This header contains three fields: date and time, calling task name, and optional short error message (provided by the calling task). The trace data are translated to displayable ASCII form and formatted for easier reading. The entire record is then written to a file. This trace file is a wrapped file (a circular buffer on secondary storage) of a fixed size.

  The second type of trace is the coprocessor or RTIC trace. The RTIC trace is similar to the PC trace in that various tasks executing on the coprocessor report their information to a centralized trace facility. However, unlike the PC trace, the RTIC trace does not have access to a file system. Therefore, the trace records are written to an 8000-byte wrapped buffer. A dump program allows the operator to write most of the important storage locations from the coprocessor (including the trace buffer) to the PC trace file.

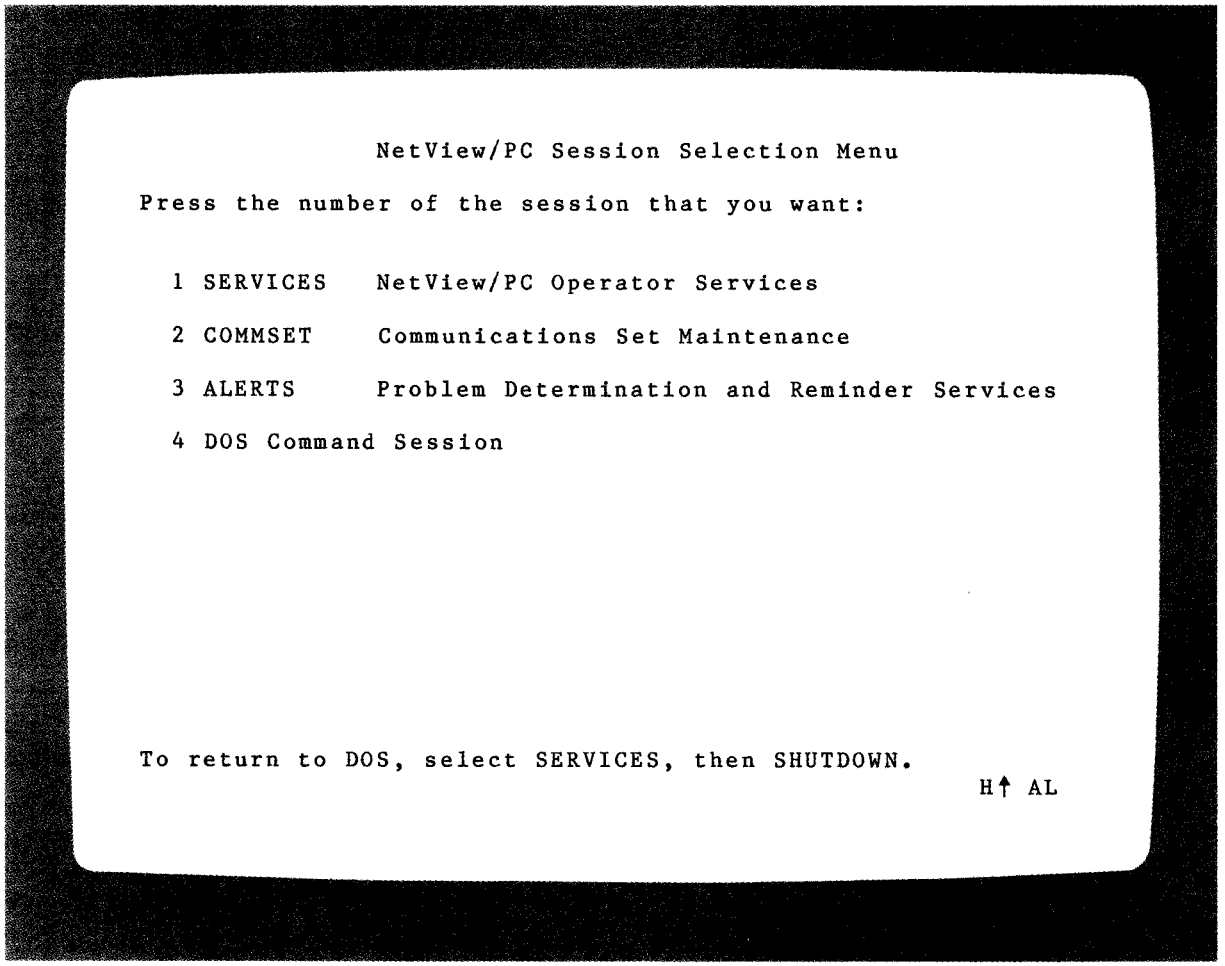
  The RTIC trace monitors the information flow between APPC and NetView/PC applications and between APPC and the host. The latter includes a line trace of all of the RUs that flow on various host sessions.

  The trace facility allows the user to perform some diagnostics when there are problems with the host communications. For example, if the NetView/PC operator cannot activate the host session because of system definition error, he/she can find the problem by looking at the trace records. This option is very useful during installation and start-up of NetView/PC.

- **Session status display**—Another usability function of NetView/PC is the session awareness function. NetView/PC displays the status of the host session on line 25 of the operator console. When the session becomes active, it displays an "H↑" icon on line 25 and sounds an audible alarm that informs the operator that the session has become active. If the session goes down due to line failure or deactivation from the host side, the operator console will display an "H↓" icon and will beep. The "H↓" flashes until the session becomes active again, or until the operator shuts down NetView/PC.

- **Automatic host session recovery**—If the host session is lost because of an abnormal condition (e.g., link failure), NetView/PC will try to recover the session without any operator intervention. This feature is useful in an environment where Net-

Figure 4 Appearance of the NetView/PC Session Selection Menu when only base applications are running



View/PC is left unattended. NetView/PC applications, such as applications for ROLM products, are written in such a way as to make use of this recovery process.

Applications

The services previously described are used by application programs. Programs running under NetView/PC can be divided into two different types according to whether they provide services primarily to a user or to another program. An application may consist of several programs of each type.

Programs with user interfaces are allocated a session, or screen group, consisting of a logical keyboard and display. A user may switch between programs via the NetView/PC session selection menu, shown in Figure 4. This menu may be accessed from any session by typing the NetView/PC hot key, which is the Scroll Lock key.

Some applications are provided with the NetView/PC base system and so are referred to as base applications. Others may be written to support specific products such as CBXs or token rings and so are referred to as product applications.

Base applications. NetView/PC applications include Operator Services, Problem Determination Services, the Host Data Facility, the Service Point Command Facility, and the Remote Console Facility.

Operator Services. The operator services application basically includes functions that are necessary but that for one reason or another are not independent applications. These functions include those listed in Table 1. Logon/Logoff is self-explanatory, as are the last three functions on the list of Table 1; the other functions are described below.

- **Intervention:** In DOS, when an error is encountered during processing of a system request (DOS function call), a message is written to the screen, and a response (if needed) is requested from the user. This normal DOS error handling cannot be used by NetView/PC applications, because NetView/PC is a multitasking system, and an error in one application cannot be allowed to interfere with the operation of others.

  When the NetView/PC program is invoked, the critical error handler in DOS is replaced by a NetView/PC error handler. If a critical error occurs, the error handler places a message on line 25 of the display (in all sessions), indicating that operator intervention is required. The user is told to select the operator intervention function in the Operator Services session. This function displays error information and allows corrective action. In addition to the DOS critical errors, some NetView/PC applications also utilize the operator intervention facility for asynchronous handling of other errors (e.g., Alert log file overwritten condition).

- **Grouping:** As discussed previously, NetView/PC contains many different applications, not all of which may be desired by any particular user at one time. In addition, a user may have multiple network management applications to be run under NetView/PC, though only one may be required at any given time, or certain ones may not be run simultaneously because of memory or other resource limitations. The Operator Services grouping function allows customization of the NetView/PC configuration. A user may select only the applications desired in a particular environment, then save the selection to a grouping file on disk. When NetView/PC is given the name of the grouping file on the command line during invocation, only the chosen applications are activated. Different grouping files may be created for different times or different operators and kept on disk.

- **Operator Records:** Another of the Operator Services is maintenance of operator records. These are records containing the name, identification, and passwords of the personnel who will be operating NetView/PC. Each record also contains a flag indicating whether the operator is the system operator; only the system operator may alter operator records.

  These records are useful when security is activated, i.e., when Logon/Logoff is required before an operator can access any NetView/PC services. In this case, the password defined in the operator records is the password accepted by the Logon/Logoff panel.

- **Region Manager:** One of the design constraints of DOS is its inability to use more than 640K of memory. This constraint limits the number of NetView/PC services that run with some applications. The Region Manager relieves this problem somewhat by providing a transient area which is shared by the Remote Console Facility, the Host Data Facility, and the Local Alert Facility. Any one of these services may be selected at one time using the Region Manager function in Operator Services. If NetView/PC is invoked with a grouping file containing a sufficiently large DOS partition, it may also be possible to bring up the Local Alert Facility in the DOS partition.

Table 1 Functions in operator services

Function	Description
Intervention	Handling of intervention messages
Grouping	Creation of grouping files
Operator Records	Operator record maintenance
Logon/Logoff	Operator log-on and log-off
Region Manager	Loading of one of the Remote Console, Host Data Transfer, or Local Alert Facilities into a transient area when selected
Host File Transfer	Transfer of a file between the local NetView/PC and a host system
Help Menu	Display of the NetView/PC General Help panels
Shutdown	Shutdown of NetView/PC

Problem Determination Services. The NetView/PC Problem Determination Services provide network error support both as a stand-alone system and as an extension of NetView network error support, i.e.,

both as a focal point and as a service point. This support is provided by a number of application programs, which offer Alert routing, Alert management, problem management, and service reminder management functions.

Alerts: Alerts are messages generated by a hardware or software product when errors are detected. Each Alert contains information regarding the type of error, the error severity, diagnostic information, the probable cause or causes, and recommended actions. In addition, the Alert contains product identification information and network routing information.

At the present time, there are two major types of Alerts, nongeneric and generic. The primary difference between these two types is that nongeneric alerts carry indices to predefined screens which are to be displayed when the Alert is viewed, and these screens are product-specific; generic Alerts carry indices to predefined text tables, which are generic to all products (hence the name generic Alerts). In addition to these two major types, there are hybrid Alerts, which carry both generic and nongeneric information. The NetView/PC Alert Facility provides a number of Alert services, including

- Routing of generic, nongeneric, and hybrid Alerts to a host system
- Local logging of nongeneric Alerts
- Local display of nongeneric Alerts
- Cross-referencing between Alerts and problems in the local database

Alerts that are sent by an application running under NetView/PC are checked for errors, then sent to the desired destination (host system, Local Alert Facility, or both). Alerts destined for the host are transformed into host format (byte reversal of integers, translation of text from ASCII to EBCDIC, etc.) before transmission. Alerts sent to the Local Alert Facility will cause an audible alarm and a visual AL icon to appear, informing the operator that a network error has occurred, unless the Alert facility has been set to ignore an Alert from that particular application.

The Local Alert Facility stores all incoming Alerts in a circular, or wrapped, history file, as well as in a pair of alternating log files which when full may be automatically off-loaded into a Device Input Format (DIF) file if desired. A list of the Alerts in the database may be obtained, sorted by various search criteria, including date and time of occurrence, resource name, application name, and so on. From the list, any Alert may be displayed, or a problem associated with a particular Alert may be created or edited.

Problems: The problem-tracking system portion of the NetView/PC Problem Determination Services allows for the creation, editing, tracking, and deletion of problem reports. These problem reports may be associated with an Alert or may be used for non-network management activities. Each problem record contains information regarding such things as the name, type, and location of a failing resource, its associated vendor, any Alerts that are associated with the problem, the date and time of a service call, present problem status, and free-form comments.

The problem database operates in the same way as the Alert database, with a circular history file, a pair of alternating log files which may be automatically off-loaded when full, and equivalent search and selection capability.

Service Reminders: In addition to error notification and problem management, the NetView/PC Problem Determination Services provides a Service Reminder Facility for scheduling routine or preventive maintenance. This facility allows the creation, editing, and tracking of service reminder records. Each record has associated with it a message and an expiration time. When the expiration time for a service reminder has passed, the operator is notified via an audible signal, and an SR icon is placed on line 25.

Reminders may be changed or reset after expiration. A list of expired reminders may also be displayed if desired, as can a list of all reminders in the database, expired or not. Reminders may be added or deleted at any time.

Host Data Facility. The Host Data Facility (HDF) allows batch (file) transfer of data between NetView/PC and CICS/DDM on a host system, using the LU 6.2 communication protocol. This function is especially useful in voice networks, where large amounts of data can be collected from various devices such as CBXs, private branch exchanges (PBXs), etc. and forwarded to a centralized facility.

As an example, the ROLM Call Detail Collector application running under NetView/PC can collect call detail information from up to 10 ROLM CBXs or some non-ROLM PBXs. This information, which is saved in a file, can later be forwarded to a host for billing or

for network analysis via the NetView/PC Host Data Facility.

HDF uses the APPC support of NetView/PC and can operate over leased or switched (dial-out) SDLC lines. The file transfer can be initiated in two ways: by an operator through the NetView/PC Host File Transfer service or by a program using an application programming interface. In either case, file transfer is performed asynchronously, allowing the operator or application to perform other tasks while the transfer is taking place.

HDF has four major functions that can be accessed through the operator panels or by any NetView/PC application through the API/CS facility. These are the SEND, RECEIVE, STATUS, and STOP functions.

SEND and RECEIVE: The SEND and RECEIVE functions allow a user to send a DOS file to or receive a file from CICS/DDM, respectively. There are various parameters and/or options that the user (operator or application) can specify with the SEND/RECEIVE functions. The TRANSLATE option will perform an ASCII-to-EBCDIC translation when sending files to the host, and will perform the reverse transformation when receiving files from the host. The checkpoint options allow the user to choose the block size for transfer of data. This option was put in for storage and performance optimization. Block sizes between 512 and 3750 bytes are allowed. For example, if the user chooses 3750 for the checkpoint size, HDF will send the file in 3750-byte increments. However, if HDF is unable to find the storage for the block size that was chosen, it will decrement the block size by 256 bytes and try to obtain the storage again. This process is repeated until the largest available storage (equal to or larger than 512 bytes) is found. HDF will inform the user of the block size that was obtained.

The larger block size will improve the performance at which the file is transferred (or received). However, because of storage constraint or line quality, the user may choose to specify a smaller block size. The OFFSET parameter allows the user to transfer data from various places within the file. This parameter defaults to zero, which means that the transfer starts at the beginning of the file.

STATUS: STATUS informs the user whether file transfer is in progress or complete. It also provides the number of bytes that have been transferred so far. HDF is an asynchronous process, which means that when the transfer has started, the user (operator or application) can continue to perform other tasks. Therefore, the user should check the status of the file transfer from time to time.

STOP: The STOP function allows the user to stop the file transfer at any point. HDF will inform the user of the number of bytes that were transferred successfully. The user may then transfer the remainder of the file at a later time, using the offset parameter discussed previously.

Service Point Command Facility. The Service Point Command Facility (SPCF) allows commands (requests) to flow from NetView to NetView/PC applications, and allows the NetView/PC applications to send replies back to NetView. These requests and replies flow in the form of a Network Management Vector Transport (NMVT) on an SSCP-PU session.

SPCF can be a powerful tool in a multivendor network environment. For example, a vendor application running on NetView/PC and monitoring a PBX can send error information to a host focal point. As a result, the operator [or a command list (CLIST)] can issue various commands to test and/or correct the problem with the device.

NetView/PC currently supports four different SPCF commands. These are LINK PD, LINK DATA, LINK TEST, and RUN.

LINK PD: This command requests an application running on NetView/PC (the service point) to perform a problem determination on a given link or link segment. The reply generated by the NetView/PC application in response to the LINK PD command contains code points that describe the results of the test. These code points are designated values that convey information to NetView regarding such items as link status (e.g., no failure detected, failure detected and isolated to resource, etc.). There are also code points for the probable cause of failure as well as the name and type of the failed resource(s).

LINK DATA: This command requests that a service point return data for a given link or link segment. The reply contains the requested data as well as the resource name and type associated with the data.

LINK TEST: This command requests that a test be performed on a link or link segment. In the reply, the service point returns the test results (i.e., pass, fail, etc.). Also included in the reply is other information such as test type, number of times the test

was performed, and a resource list which contains all the resource names pertaining to this test.

RUN: This command contains user-defined text, which is used by the service point application to carry out various functions. The reply also contains user-defined text, which is displayed at the NetView console. Optionally a CLIST can be triggered by this reply, causing other commands to be sent by the focal point.

The Remote Console Facility allows one NetView/PC to monitor or control another NetView/PC.

Remote Console. Another tool provided for network management by NetView/PC is the Remote Console Facility. This facility is designed to enhance the productivity of network management personnel by allowing remote problem determination. The Remote Console allows personnel at one NetView/PC to have an operator session with a NetView/PC at another location over a communications link such as a phone line. The remote operator can either control or monitor the operation of the target (or local) NetView/PC and also initiate file transfer between the two stations. When the Remote Console Facility is active, an RC icon is displayed on line 25 in all sessions.

File Transfer: In cases where a memory dump or other diagnostic information has been written to a disk file, it may be necessary to transfer the file to another location for analysis. The Remote Console Facility has a file transfer capability which allows files to be transmitted in both directions over a communications line between a local and a remote NetView/PC installation. This provides a means of distributing diagnostic information. It may also be used for transferring other information; for example, a central NetView/PC may call NetView/PC installations that are attached to different CBXs and retrieve the file containing call detail records from these stations.

Control: In cases where more interactive problem determination is required, the Remote Console Facility allows the operations of the NetView/PC running on a target (local) machine to be controlled by an operator at a remote NetView/PC; the same communications line utilized for file transfer is used. When in this mode, the local keyboard is locked while keystrokes typed at the remote keyboard are passed on to the local NetView/PC as if they were being typed at the local keyboard. The display on the local machine is duplicated at the remote NetView/PC in order to provide feedback.

Monitoring: The Remote Console Facility also allows a remote NetView/PC to monitor the local NetView/PC without having control. In this mode, a local operator can demonstrate a problem to an operator at a central site or to a service representative. The local operator is in control of the local NetView/PC, and the information on the display is duplicated at the remote NetView/PC.

The local operator allows access by the remote operator in control or monitor modes and can switch between them at will.

Users are able to switch between control and monitoring modes at will. In either mode, a log-on is required if the security option in the NetView/PC Operator Services were selected.

Product applications. A major function of NetView/PC is to act as a base system to application programs that manage physical devices. These programs may be either IBM or vendor applications.

IBM applications. A number of IBM applications run under NetView/PC, including the IBM Token-Ring Manager, the ROLM Alert Monitor, and the ROLM Call Detail Collector.

IBM Token-Ring Manager: The Token-Ring Manager is a network management program for the IBM token ring. The Manager provides problem determination and error recovery services for one or more rings. It operates either as a stand-alone program or as an application under NetView/PC.

One of the major functions of the Token-Ring Manager is to monitor the ring for hard and soft errors, "soft" generally indicating transient or recoverable errors and "hard" indicating faults in the token-ring cabling system, access units, or network adapters. If an Alert condition occurs, an Alert is sent to Net-

View/PC, which provides both an audible and visual indication of an Alert condition. The Alert may also be sent to NetView simultaneously, using the Alert facility and host communications services of NetView/PC.

ROLM CBX applications: NetView/PC, in conjunction with the ROLM Alert Monitor and Call Detail Collector application programs, integrates voice and data management into IBM's network management architecture.

The Alert Monitor assists in problem notification and determination of ROLM Computerized Branch Exchanges (CBXs). Error and alarm conditions detected by a CBX are encoded in Alerts and forwarded to NetView/PC and, if desired, to NetView. The Call Detail Collector functions in the same manner. The primary difference is that the types of data collected are call detail records from CBXs or pollable storage devices, where the storage devices are connected to remote CBXs or to some non-ROLM PBXs.

Information may be transferred from the CBXs or pollable storage devices via a dedicated asynchronous connection or through a dial connection initiated either by NetView/PC or the storage device. If a dial line is used, connections can be initiated by NetView/PC at regularly scheduled intervals.

These two applications, running under NetView/PC, collect error and call-detail information from one or more CBXs to be processed locally and/or sent to a central site for further processing.

Vendor applications. One of the major attractions of NetView/PC is that non-IBM vendors are offered facilities which allow their applications to run under NetView/PC. This enables vendor products to be integrated into an SNA network management control point, thus benefiting the customer by allowing one centralized control facility to manage an entire network. Two NetView/PC features were specifically developed for vendor applications: the DOS Partition and the Open API.

DOS Partition: The IBM-developed operating system under which NetView/PC runs on top of DOS did not provide facilities by which vendors could specify that their applications were to be started along with other NetView/PC applications. Because of this, and in order to improve usability by allowing a user to use some DOS programs while running NetView/PC, the DOS Partition was developed. The DOS Partition is basically a session that runs DOS (a secondary COMMAND.COM) as its name implies. It is selectable from the NetView/PC session selection menu. Any application that is well-behaved may be run in the DOS Partition.

The term well-behaved means that the program obeys the following rules, among others:

1. Does not do direct writes to the video buffer
2. Does not execute a terminate-and-stay-resident call
3. Does not destroy DOS memory management information
4. Does not take over DOS or BIOS interrupts
5. Does not take over the timer interrupt
6. Does not take over the critical error interrupt
7. Any interrupts, other than the above, that are taken over by the application, must be restored upon exit

Some examples of DOS programs that will work in the DOS Partition are FORMAT and TREE. Programs which were tested and found to work are the IBM Personal and Professional Editors, VisiCalc®, the IBM Macro Assembler, and the Computer Innovations C86 C compiler.

A number of software vendors have developed or are in the process of developing applications that run in the DOS Partition under NetView/PC.

Open API: NetView/PC is written in an IBM proprietary development language. The interfaces provided to NetView/PC base applications are generally sets of macros. Obviously, this type of interface is not accessible to non-IBM applications because it requires an IBM-specific language and because it requires access to the NetView/PC development library at compile time. In order to allow vendors access to NetView/PC functions, an open application programming interface (Open API) was provided. The interface enables vendors to use many of the NetView/PC services.

This API basically consists of a number of DOS OBJ (object) modules which are linked to an application. The application calls the routines in these modules, which act as a gateway between the application and the NetView/PC services.

A number of NetView/PC functions have been made available through the API at this time. There are four major components of the API. The Alert API allows

a program to send an Alert to a host or to the Local Alert Facility. The SPCF API allows a program to receive and parse commands from a host system as well as to build and send replies back to the host. The operator communications API places an icon on line 25 to indicate that the application running in the DOS Partition requires operator attention. The HDT/API allows transfer of files between the IBM PC and a host system.

These functions assist in integrating vendor applications into the NetView/PC system.

Summary

NetView/PC is a product with a great deal of functionality. It provides service point functions to enable non-SNA devices to participate in an SNA network management environment. It also offers focal point services which allow NetView/PC to be used for network management in a non-SNA environment. The multitasking environment and an open API allow both IBM and non-IBM products to develop applications which run under NetView/PC. Thus, customers can have greater centralization of network management.

General references

Systems Network Architecture Format and Protocol Reference Manual, SC30-3346, IBM Corporation (June 1987); available through IBM branch offices.

Systems Network Architecture Formats, GA27-3136, IBM Corporation (June 1987); available through IBM branch offices.

M. E. Sprague, *NetView/PC Application Programming Interface/ Communications Services,* SC30-3313, IBM Corporation; available through IBM branch offices.

R. J. Sundstrom, J. B. Staton III, G. D. Schultz, M. L. Hess, G. A. Deaton, Jr., L. J. Cole, and R. M. Amy, "SNA: Current requirements and direction," *IBM Systems Journal* **26,** No. 1, 13–36 (1987).

D. H. Thoenen, *IBM/ROLM Voice Network Management,* GG22-9120, IBM Corporation (September 1986); available through IBM branch offices.

Mehrdad "Mo" Ahmadi *IBM Communication Products Division, P.O. Box 12195, Research Triangle Park, North Carolina 27709.* Mr. Ahmadi joined IBM in 1982, working on the development of the Virtual Telecommunications Access Method (VTAM). He later joined the Network Control Program (NCP) performance test group. Mr. Ahmadi is currently a staff programmer in the NetView/PC communications department, and has been responsible for the design and development of portions of the host communication support of NetView/PC.

James H. Chou *IBM Communication Products Division, P.O. Box 12195, Research Triangle Park, North Carolina 27709.* Mr. Chou joined IBM in 1984. He has worked with the Link Level Control test group for the IBM Token-Ring adapter card for the IBM PC. He is currently an associate programmer in the NetView/PC User Interface department, and is working on enhancing NetView/PC focal point functions. Mr. Chou received a B.S. in biology and a B.S. in electrical engineering from the Massachusetts Institute of Technology in 1983. He is at present a candidate for the Master's and Ph.D. degrees in computer science at the University of North Carolina at Chapel Hill.

Gerry Gafka *IBM Communication Products Division, P.O. Box 12195, Research Triangle Park, North Carolina 27709.* Mr. Gafka is a senior programmer manager responsible for the NetView/PC user interface. He was responsible for developing the Vendor Feedback Program during the initial NetView/PC development. The IBM Token-Ring LAN and Line Switching development activities are recent areas in which he has made contributions and held management positions. Mr. Gafka has previously managed groups in IBM's Kingston, New York, facility, where he contributed to the 8100 DPPX intercomponent interface design, release planning, system build, and library control. Mr. Gafka has also held technical and management positions at the IBM facility in Poughkeepsie, New York. The areas in which he worked were OS/360 design, development, system testing, programming service support, system release, and programming quality.

Reprint Order No. G321-5309.